



Bruno Miguel
Marques da Rocha

Leitura autónoma de contadores de água com
deteção de padrões de consumo e fugas



**Bruno Miguel
Marques da Rocha**

**Leitura autónoma de contadores de água com
deteção de padrões de consumo e fugas**

Dissertação apresentada à Universidade de Aveiro, para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Automação Industrial, realizada sob a orientação científica do Doutor Paulo Bacelar Reis Pedreiras, Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro, e sob a co-orientação do Doutor Valter Filipe Miranda Castelão da Silva, Professor Adjunto da Escola Superior de Tecnologia e Gestão de Águeda.

o júri / the jury

presidente / president

Professor Doutor Pedro Nicolau Faria da Fonseca

Professor Auxiliar da Universidade de Aveiro

vogais / committee

Professor Doutor Frederico Miguel do Céu Marques dos Santos

Professor Adjunto do Instituto Superior de Engenharia de Coimbra (arguente)

Professor Doutor Valter Filipe Miranda Castelão da Silva

Professor Adjunto da Universidade de Aveiro (Co-orientador)

agradecimentos / acknowledgements

Em primeiro lugar gostaria de agradecer aos meus orientadores.

O professor Doutor Paulo Pedreiras, pela paciência, partilha de conhecimento ao longo de todo o trabalho, e motivação que transmitiu revelando-se fundamental para conseguir superar dificuldades encontradas.

Ao professor Doutor Valter Silva, por toda a paciência, motivação e disponibilidade demonstrada. Pelo excelente trabalho que faz, para entender as dificuldade dos seus alunos, e arranjar estratégias para as superar.

Quero também agradecer ao professor Doutor. Vitor Santos, que de uma forma indireta me ajudou imenso, por disponibilizar o Laboratório de Automação e Robótica para trabalhar e pelo empenho demonstrado nas aulas de sistemas de visão. Conhecimentos, que se revelaram fundamentais no desenvolvimento deste trabalho. Quero agradecer aos amigos da ESTGA, pela apoio que sempre me deram ao longo destes anos! Em especial ao Engenheiro Mauro Antunes, pelas incontáveis vezes que trabalhamos juntos, não só este ano, mas em todos os projetos em que já participamos. Com a partilha de conhecimento, sempre foi possível superar imensas dificuldades ao longo destes anos.

Aos meus pais, irmão e irmã, pela disponibilidade em apoiar e discutir ideias, que ajudaram a melhorar o trabalho desenvolvido.

Agradeço também aos meus amigos, que sempre se mostraram dispostos em falar sobre o progresso da dissertação. Em especial, à minha amiga Betty Marques pelo apoio demonstrado e ao Diogo Carvalho, que desde os tempos de infância, sempre partilhou imenso conhecimento.

Aos meus amigos Engenheiros Bruno Carvalho Vieira e Luís Emanuel Cristóvão, pelas as ideias discutidas, e partilhas de material ao longo deste ano, que me ajudaram imenso nesta jornada!

Agradeço também, à malta do MEAI e do DEM, que sempre se mostraram disponíveis para ajudar e partilhar conhecimento.

palavras-chave

Contador de água, *Retrofitting*, Fugas, Contadores analógicos, Redes de abastecimento

resumo

A água é um bem escasso e essencial para a saúde pública. Em Portugal, sensivelmente 30% da água é perdida nas redes de abastecimento, que representam custos anuais de 280 milhões de euros. As entidades responsáveis pelo fornecimento de água necessitam de controlar e monitorizar as redes de abastecimento, com o propósito de evitar perdas avultadas.

Para proceder à monitorização, os contadores devem encontrar-se equipados com tecnologias de comunicação, que permitam enviar os consumos medidos para a entidade responsável pelo serviço. Os contadores inteligentes, são equipamentos que possibilitam a transmissão dos consumos, todavia, o custo pode variar entre 200 e 250 euros o que impede que as empresas os apliquem massivamente.

Já os contadores analógicos e digitais convencionais, são equipamentos mais antigos e não permitem a transmissão dos consumos medidos. No entanto, encontram-se instalados em elevado número nas redes de abastecimento.

O propósito desta dissertação, é desenvolver um sistema que permita reaproveitar os contadores convencionais, de modo a enviarem os seus consumos e analisar os dados lidos para detetar possíveis fugas de água.

Na pesquisa efetuada, vários indicadores apontaram para desenvolver um sistema com base em sistemas de visão que permitam ler os consumos no contador. Esta técnica, tem a vantagem de necessitar meramente da câmara e sistema de iluminação.

O sistema desenvolvido é composto pelo leitor, responsável por ler o valor no contador através de sistemas de visão, validar o resultado da leitura, guardar os dados, analisar possíveis fugas na habitação e enviar os consumos para um servidor numa data agendada.

O servidor tem a finalidade de receber os consumos efetuados pelos leitores instalados, guardar os dados e analisar se existem fugas através das diferenças dos valores lidos e emitir alarmes ao utilizador.

Para realização dos testes ao sistema, construiu-se um prototipo que equipa um contador analógico, e realizaram-se diversas abordagens de modo a aumentar a taxa de acerto final. Estas permitiram subir a taxa de acerto de 92% com 1000 imagens, para 100% em 8000 imagens do contador. Este resultado deve-se à aplicação das ferramentas na análise de imagem e validação dos valores lidos pelo algoritmo *OCR*.

Os ensaios de deteção de fugas em habitações e em redes de abastecimento, foram simulados com valores lidos do prototipo. Foram realizados ensaios relativos à deteção de fugas, quer no que respeita a habitações quer à rede pública, tendo sido verificado que o sistema foi capaz de sinalizar os padrões anormais de consumo bem como as perdas de água na rede inseridos como parte dos procedimentos de validação.

Relativamente aos tempos de execução do código, se este for executado numa placa de desenvolvimento *Raspberry Pi 3*, demoraria aproximadamente 2 por minuto, o que possibilitaria fazer acima de 86400 leituras por mês.

keywords

Water counter, *Retrofitting*, Leaks, Analog counters, Supply networks

abstract

Water is scarce and essential to public health. In Portugal, approximately 30 % of the water is lost in the supply networks, which represent annual costs of 280 million euros. The entities responsible for water supply need to control and monitor the supply networks, in order to avoid large losses.

In order to carry out the monitoring, the meters must be equipped with communication technologies, which allow the measured consumption to be sent to the entity responsible for the service. Smart meters are devices that allow the transmission of consumption, however, its cost can vary between 200 and 250 euros which prevents companies from applying them massively. Conventional analog and digital meters are older equipment and do not allow the transmission of measured consumption, however, they are installed in large numbers in the supply networks.

The purpose of this dissertation is to develop a system that allows reusing the analog meters to send their consumption and analyze the data read to detect possible water leaks. In the research carried out, several indicators pointed to the development of a system based on vision that allow to read the consumptions in the counter. This technique has the advantage of merely needing a camera and a lighting system.

The system developed is composed of reader, responsible for a reading the non-counter value through vision systems, validate the reading result, save the data, analyze possible leaks in the house and send the consumptions to a server in a scheduled data.

The server has a purpose of receiving the readings from the installed players, saving the data and analyzing if there are leaks based on the difference between the values read.

For the realization of the tests to the system, a prototype was constructed that works on an analog counter, and diverse approaches were realized in order to increase the final adjustment rate. These allowed to increase the hit rate of 92 % with 1000 images, to 100 % in 8000 counter images. This result is due to the application of the tools in the image analysis and validation of the values read by the OCR algorithm. The tests of leakage detection in dwellings and in supply networks were simulated with values read from the prototype.

In the leakage testing in dwellings, a limit of $0.1m^3/min$ was set and there were 37 passages above the ceiling, for a total of $17.3m^3$ of wasted water. Regarding leak detection testing in the supply networks, losses of $1127.5m^3$ of water were detected.

Regarding the code execution times, if it runs on a Raspberry Pi 3 development board, it would take approximately 2 by minute, and would make it possible to make over 86400 readings per month.

Conteúdo

Conteúdo	i
Lista de Figuras	v
Lista de Tabelas	vii
Acrónimos	ix
1 Contexto e problema a resolver	1
1.1 Objetivos	2
1.2 Organização da dissertação	2
2 Estado de arte	3
2.1 Contador analógico e digital	3
2.2 Contador inteligente	4
2.3 Implementação de tecnologias ao contador analógico	4
2.4 Leitura em contadores analógicos através de sistemas de visão	5
2.5 Algoritmos de processamento de imagem	6
2.6 Reconhecimento aplicado à imagem	6
2.7 Comparação entre <i>Tesseract</i> e <i>FineReader</i>	8
2.8 Pré-processamento e reconhecimento aplicado ao contador	9
2.8.1 Leitura dos algarismos em movimento	10
2.9 Soluções comerciais existentes	12
2.9.1 Equipamento <i>Metlink</i>	12
2.9.2 Equipamento <i>RS2000</i>	12
2.9.3 Equipamento <i>DD3G</i>	13
2.9.4 Equipamento <i>B280 prime</i>	13
2.9.5 Resumo sobre os equipamentos analisados	14
2.10 Rede de contadores ligados através da <i>Internet</i>	14
2.11 Detecção de fugas nas redes de distribuição de água	15
2.12 Identificação de fugas em habitações particulares	16
2.13 Conclusão do estado de arte	17
3 Solução de <i>hardware</i> proposto	19
3.1 Requisitos para desenvolvimento do produto	19

3.2	Estrutura da unidade de leitura	20
3.2.1	Unidade de processamento	20
3.2.2	Resumo sobre placas pesquisadas	22
3.2.3	Módulo de comunicação	23
3.2.4	Câmara de vídeo	23
3.2.5	Iluminação artificial	24
3.2.6	Resultado da montagem do <i>Hardware</i>	25
4	Solução de software proposto	27
4.1	Requisitos do software	27
4.2	Introdução ao Back End	28
4.3	Principais funções do Back End	28
4.4	Introdução ao <i>software</i> aplicado ao CA	29
4.5	Configuração da aplicação para leitura autónoma	30
4.5.1	Seleção da <i>Region of Interest</i>	30
4.5.2	Binarização da imagem	32
4.5.3	Aplicação do <i>BlobsCleaner</i>	32
4.6	Verificação da posição dos algarismos no contador	33
4.6.1	Nível de visibilidade e acerto final	37
4.7	Verificação do valor lido	38
4.8	Deteção de fugas	39
4.9	Fluxograma do sistema em modo autónomo	43
4.9.1	Gestão de dados em modo autónomo	43
5	Testes e validações	45
5.1	Teste ao sistema de leitura	45
5.1.1	Ensaio ao sistema de visão 1	46
5.1.2	Ensaio ao sistema de visão 2	48
5.1.3	Ensaio ao sistema de visão 3	49
5.1.4	Ensaio ao sistema de visão 4	51
5.2	Simulação de fuga numa habitação	53
5.3	Simulação de fuga em rede de abastecimento	56
5.4	Tempo de ciclo do programa	58
5.4.1	Tempo de execução de código	58
5.4.2	Tempo de aquisição de imagem pela câmara	59
5.5	Tempo relativo de execução do código, se aplicado à placa de desenvolvimento <i>Raspberry Pi 3</i>	59
6	Conclusão e trabalho futuro	61
	Bibliografia	63
A	Principais aplicações instaladas	65
B	Lista das principais funções desenvolvidas no Cliente	66

C	Lista das principais funções desenvolvidas no Servidor	68
D	Imagens da Interface gráfica	70

Lista de Figuras

1.1	(a) Contador analógico de água [2] (b) Contador analógico de energia elétrica [3]	1
2.1	(a) Contador analógico de água [9] (b) Contador digital de água [10]	3
2.2	Contador digital inteligente [11]	4
2.3	(a) Contador analógico alterado com tecnologias de <i>encoder</i> [12] (b) Contador analógico alterado com tecnologias de pulso magnético [13]	5
2.4	Contador digital inteligente [14]	5
2.5	Sistema de transmissão dados através do uso de <i>smartphone</i> [15]	6
2.6	Principais processos para reconhecimento de imagem [16]	7
2.7	Distância e condições de captura da imagem [16]	7
2.8	Processo geral de leitura [4]	8
2.9	Imagem original obtida pela câmara de vídeo	10
2.10	Pré-processamento da imagem frontal obtida pela câmara de vídeo	10
2.11	Contador analógico e algarismo em posição indefinida	10
2.12	Visibilidade dos algarismos na imagem pré-tratada	11
2.13	Fluxograma do algoritmo de leitura [16]	11
2.14	Sistema <i>Metlink</i> [18]	12
2.15	Equipamento <i>RS2000</i> [19]	13
2.16	Equipamento <i>DD3G</i> [20]	13
2.17	Equipamento <i>B280 Prime</i> [21]	14
2.18	Rede de ligação adaptada de [14]	15
2.19	Conjunto de contadores aplicados à rede de abastecimento	16
2.20	Rede de abastecimento de uma habitação	17
3.1	Estrutura do modelo desenvolvido	20
3.2	<i>Xbox Live Vision</i> [25]	24
3.3	Conjunto de <i>LEDs</i> [24]	24
3.4	(A) Câmara de vídeo (B) Pilha de alimentação para a iluminação artificial aco- plada ao encapsulamento (C) Base onde se encontra o motor para movimentar o contador	25
3.5	(a) Parte superior, com iluminação e câmara de vídeo (b) Parte inferior, com o contador	26
4.1	Esquema completo do sistema	28
4.2	Fluxograma do sistema de leitura e envio de dados para o servidor	30

4.3	Imagem do contador sem seleção de área para leitura	31
4.4	Imagem do contador com seleção de área para leitura	31
4.5	Imagem que irá ser sujeita aos tratamentos de imagem	31
4.6	(a) <i>Threshold</i> a nível 46 (b) <i>Threshold</i> a nível 117	32
4.7	<i>Threshold</i> a nível 85	32
4.8	(a) <i>BlobsCleaner</i> a nível 0 (zero) (b) <i>BlobsCleaner</i> a nível 2000	32
4.9	<i>BlobsCleaner</i> aplicado a nível 845	33
4.10	Imagem do contador com aplicação do <i>Threshold</i>	33
4.11	Imagem vinda da aplicação do <i>BlobsCleaner</i>	33
4.12	Imagem final após aplicação do <i>BlobsCleaner</i>	33
4.13	Contadores (a) e (b) em transição	34
4.14	Aplicado tratamento as figuras no exemplo (a) e (b)	34
4.15	Resultados obtidos das leituras do exemplo (a) e (b)	34
4.16	Modo de deteção no exemplo (a) e (b)	35
4.17	Imagem final do exemplo (a) e (b)	35
4.18	Contador com transição quase completa	36
4.19	(a) Imagem pré-tratada (b) Imagem final adquirida	36
4.20	Imagem do contador com aplicação de <i>Threshold</i>	36
4.21	Percentagem de acerto e visibilidade	38
4.22	Fluxograma de verificação do valor lido pelo algoritmo	38
4.23	Fluxograma geral para a deteção de fugas no leitor	41
4.24	Fluxograma geral para deteção de fugas no servidor	42
4.25	Relação entre os contadores instalados na rede de abastecimento	42
4.26	Fluxograma do algoritmo em modo autónomo	43
5.1	Leituras desde 0 até 3000 no contador em 1276 imagens	47
5.2	Diferenças entre os números lidos do ensaio 1	47
5.3	Leituras desde 0 até 3000 no contador em 1276 imagens	48
5.4	Diferenças entre os números lidos do ensaio 2	49
5.5	Leituras desde 57452 até 57709 em 920 imagens	50
5.6	Diferenças entre os números lidos no ensaio 3	51
5.7	Leituras desde 50031 até 56113 em 8000 imagens	52
5.8	Diferenças entre os números lidos no ensaio 4	53
5.9	Leituras desde 57109 até 57651 em 906 imagens	54
5.10	Diferenças entre os números lidos da simulação	54
5.11	Consumos registados acima do limite definido	55
5.12	Simulação de fuga na rede de abastecimento	56
5.13	Leitura estimada e medida	57
5.14	Diferenças entre os valores estimado e medido	57
D.1	Menu principal Servidor	70
D.2	Funcionalidade para adicional dispositivos	70
D.3	Funcionalidade para verificar as características	71

Lista de Tabelas

2.1	Comparação entre os algoritmos <i>OCR FineReader</i> e <i>Tesseract</i> adaptada de [16]	9
2.2	Resumo dos produtos do mercado comercial	14
3.1	Comparação dos equipamentos da empresa <i>Raspberry Pi</i>	21
3.2	Comparação dos equipamentos da empresa <i>Arduino</i>	21
3.3	Comparação dos equipamentos da empresa <i>Orange</i>	22
3.4	Comparação dos equipamentos <i>Asus</i> e <i>MediaTek</i>	22
3.5	Principais especificações do equipamento <i>SIM800H</i>	23
3.6	Principais especificações da câmara <i>Microsoft Xbox 360 Live Vision</i>	24
4.1	Funcionalidade adicionar dispositivos	29
4.2	Funcionalidade consulta de dados	29
4.3	Funcionalidade alerta de fugas	29
4.4	Configuração inicial aplicada pelo utilizador no ensaio 1	37
4.5	Dados de pré-configuração em modo autónomo	44
4.6	Dados enviados do leitor para o <i>Back-end</i>	44
5.1	Tipo de alterações executadas no código ao longo dos ensaios	46
5.2	Configuração inicial aplicada pelo utilizador no ensaio 1	46
5.3	Alterações feitas ao sistema no ensaio 1	46
5.4	Resultados do ensaio 1	48
5.5	Alterações feitas ao sistema do ensaio 2	48
5.6	Resultados do ensaio 2	49
5.7	Configuração inicial aplicada pelo utilizador no ensaio 3	49
5.8	Alterações feitas ao sistema no ensaio 3	50
5.9	Resultados do ensaio 3	51
5.10	Configuração inicial aplicada pelo utilizador no ensaio 4	52
5.11	Alterações feitas ao sistema no ensaio 4	52
5.12	Resultados do ensaio 4	53
5.13	Resultados da simulação	55
5.14	Dados sobre os consumos no ensaio	55
5.15	Resultados da análise do teste do sistema	55
5.16	Resultados da análise do teste do sistema	58
5.17	Dados de teste do sistema	58
5.18	Ensaio ao teste de velocidade feito a cinco imagens diferentes	58

5.19 Média total, máximo e mínimo no tempo de execução	59
A.1 Principais bibliotecas instaladas	65
B.1 Principais funções desenvolvidas	67
C.1 Principais funções desenvolvidas	69

Acrónimos

<i>3G</i>	Third Generation
<i>CA</i>	Contador Analógico
<i>CI</i>	Contador Inteligente
<i>CPU</i>	Central Processing Unit
<i>EDGE</i>	Enhanced Data GSM Environment
<i>GIP</i>	Glade Interface Designer
<i>GPIO</i>	General Purpose Input/Output
<i>GPRS</i>	General Packet Radio Services
<i>GPS</i>	Global Positioning System
<i>GSM</i>	Global System for Mobile Communication
<i>HDMI</i>	Hight Definition Multimedia Interface
<i>IDE</i>	Integrated Development Environment
<i>IP</i>	Internet Protocol
<i>LCD</i>	Liquid Crystal Display
<i>MicroSD</i>	Micro Secure Digital
<i>OCR</i>	Optical Character Recognition
<i>RAM</i>	Random Access Memory
<i>ROI</i>	Region Of Interest
<i>RS232</i>	Recommended Standard 232
<i>SIM</i>	Subscriber Identify Module
<i>SMS</i>	Short Message Service

Capítulo 1

Contexto e problema a resolver

A água é bem escasso e essencial para manter a saúde pública de um país. Em Portugal continental, as perdas de água podem chegar a 30%. Destas, 20% correspondem a perdas em tubagens (fugas) e os restantes 10% a consumos autorizados, mas não faturados, que podem ocorrer por exemplo no abastecimento dos bombeiros em acontecimentos de calamidade pública. A água total distribuída, corresponde a 820 milhões de metros cúbicos, que cerca de 280 milhões não são faturados, representando perdas anuais acima de 160 milhões de euros [1].

As entidades que têm ao seu encargo a distribuição de água, gás ou eletricidade necessitam, para faturação, executar leituras relativas aos consumos dos seus clientes. No caso dos contadores analógicos (CA), a realização dessas leituras implica o deslocamento de um funcionário ao local onde o equipamento está instalado, para efetuar o registo dos consumos. A figura 1.1 mostra dois exemplos desses mesmos contadores.



Figura 1.1: (a) Contador analógico de água [2] (b) Contador analógico de energia elétrica [3]

Este método de registo é demorado e oneroso, razão pela qual as empresas fornecedoras dos serviços como por exemplo de energia elétrica, optam por emitir faturas mensais com valores estimados, efetuando leituras reais mais espaçadamente [4].

As faturas de valor estimado têm um erro relativo ao consumo real, logo, é conveniente desenvolver um sistema automático, que permita efetuar leituras remotamente, e transferi-las para uma base de dados na empresa fornecedora do serviço, a um custo reduzido.

Pela análise de dados é possível adicionar funcionalidades, que permitam a deteção de fugas pela observação de mudanças drásticas nos consumos [5].

Um sistema deste género com capacidade de transmitir os consumos através de tecnologias de comunicação sem fios, permite evitar deslocações periódicas ao local, e possibilita que haja várias leituras do contador durante o período de faturação.

Os contadores inteligentes (CI), possibilitam a transmissão dos consumos para as empresas fornecedoras de serviços. No entanto, a instalação destes obriga à remoção do antigo contador analógico convencional. Embora haja variação no custo dos equipamentos, o preço médio dos contadores inteligentes oscila entre 200 e 250 euros por cliente [7]. Estes custos limitam fortemente a sua aplicação, pelo que neste trabalho é proposto um equipamento de baixo custo para a massificação de leituras mantendo os contadores analógicos convencionais [6].

A diretiva 2009/73/CE estabelece a obrigatoriedade dos Estados-Membros procederem à implementação de sistemas com contadores inteligentes de eletricidade e gás natural. Com base nesta diretiva, a autoridade deve fixar um calendário correspondente a um período de 10 anos com vista à implementação destes contadores. Este desenvolvimento representa 200 milhões de novos contadores inteligentes para medição de eletricidade e 45 milhões para gás natural, até 2020. O investimento está avaliado em 45 milhões de euros e espera-se que em 2020, 70% dos consumidores Europeus estejam equipados com contadores inteligentes de eletricidade e 40% para gás natural [8].

1.1 Objetivos

A proposta consiste em aproveitar os contadores convencionais instalados nas redes de abastecimento, através do desenvolvimento de um equipamento que permita executar as leituras autonomamente.

Embora haja contadores que permitem medir consumos de água, gás e eletricidade, nesta dissertação o trabalho incidirá unicamente sobre os contadores analógicos de água.

O equipamento que se pretende desenvolver, espera-se constituído por componentes de baixo custo e que seja aplicável a diversos géneros de contadores convencionais. Deve igualmente ter a capacidade de detetar possíveis fugas nas redes de abastecimento, pela análise dos consumos efetuados no contador. Pretende-se que seja de fácil implementação e não necessite mão de obra especializada.

Por último, o sistema deve funcionar de forma autónoma e ter capacidade para comunicar através de tecnologias sem fios. Os dados adquiridos devem ser enviados para um servidor que é responsável pela gestão e controlo dos consumos.

O servidor deve disponibilizar novos serviços, por exemplo, capacidade para alertar o consumidor sobre alarmes recebidos dos leitores instalados na rede de abastecimento.

1.2 Organização da dissertação

Esta dissertação está dividida em seis capítulos. O primeiro refere-se à problemática com as perdas de água. O seguinte apresenta os contadores existentes, e a revisão dos resultados académicos e produtos industriais mais relevantes, definindo também os requisitos necessários.

O terceiro e quarto capítulos apresentam a solução de *hardware* e *software* proposta. No quinto capítulo apresenta os testes ao *hardware* e *software* desenvolvidos e quais os resultados práticos.

Por último, o sexto capítulo aborda as conclusões, reflexões sobre os resultados propondo trabalho futuro para complementar o sistema desenvolvido.

Capítulo 2

Estado de arte

Este capítulo, destina-se a introduzir os principais géneros de contadores existentes. Pretende igualmente analisar-se os trabalhos a nível académico e comercial desenvolvidos.

A literatura estudada tem origem em documentação académica e comercial. Este estudo, destina-se a perceber em que ponto de situação e qual o rumo que a comunidade científica está a impulsionar no desenvolvimento da área, e quais os equipamentos aplicados em ambientes industriais.

2.1 Contador analógico e digital

Os primeiros contadores analógicos instalados remontam ao ano de 1880. Desde essa altura, têm aumentado a sua aplicação tanto em empresas como casas particulares, com a finalidade de medir variáveis como a água, gás ou eletricidade. Os proprietários destes equipamentos, usualmente usam-nos para cobrar dividas aos clientes pelos consumos efetuados.

Os primeiros contadores, foram desenvolvidos com tecnologias mecânicas, na construção usam-se anéis alinhados e enumerados de 0 a 9, onde através deles é possível ler os consumos medidos. Na figura 2.1 (a), está representado o contador analógico descrito.

Posteriormente com o avanço da tecnologia, foi desenvolvido o contador digital. Este equipamento, apresenta os consumos efetuados ao utilizador através de um ecrã LCD. Estes contadores, podem ser programados de forma a alterar os horários de consumo dos seus utilizadores. A figura 2.1 (b) apresenta o contador digital.

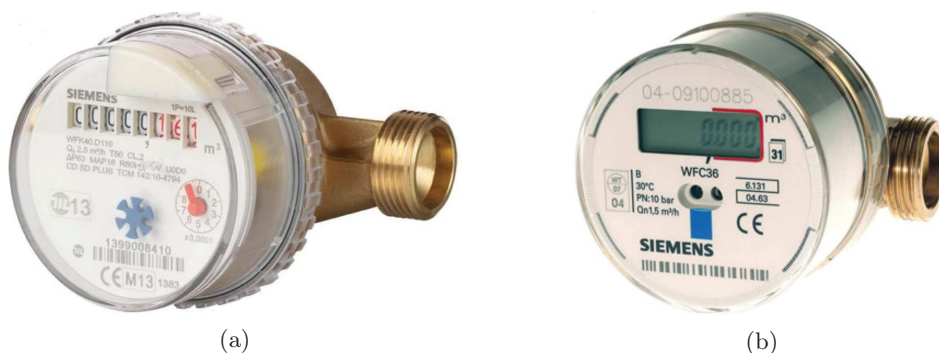


Figura 2.1: (a) Contador analógico de água [9] (b) Contador digital de água [10]

O contador analógico e o digital, têm uma inferioridade em comum, ambos não possuem

tecnologias que permita transmitir os consumos efetuados. Para colmatar esta falha, foram posteriormente desenvolvidos os contadores inteligentes.

2.2 Contador inteligente

Um contador inteligente ou *smart meter*, é um equipamento elétrico com capacidade de transmitir os consumos medidos, através de sistemas de comunicação sem fios. Estes dados são enviados para as empresas responsáveis, que posteriormente cobram os consumos aos seus clientes. Na figura 2.2, está representado um contador inteligente.



Figura 2.2: Contador digital inteligente [11]

Os consumos efetuados são enviados para uma central e podem ser registados em intervalos variáveis. Estes dados, são posteriormente analisados por ferramentas, que permitem detetar fugas nos sistemas de distribuição.

Todavia, estes equipamentos têm custos que podem variar entre 200 e 250 euros, e fazem com que as empresas não os apliquem massivamente.

O *retrofitting*, é uma prática que permite aplicar novas tecnologias aos contadores analógicos e digitais, ou seja, fazer o seu aproveitamento. Este processo, permite modernizar os contadores e trazer benefícios, que são explicados na secção 2.3 seguinte. No entanto, nesta dissertação só é focado a aplicação de novas tecnologias aos contadores analógicos, mas é perfeitamente adaptável aos digitais.

2.3 Implementação de tecnologias ao contador analógico

O *retrofitting* de um equipamento, permite aprimorar características, que antes eram consideradas obsoletas. Este processo aplicado aos contadores analógicos, permite que os dados sejam transmitidos através de redes sem fios.

Na figura 2.3 (a), o contador foi alterado e aplicado um *encoder*, que permite medir a posição dos ponteiros e assim determinar o consumo efetuado.

No segundo exemplo (b), é apresentada uma solução baseada em contagem por pulsos. Para implementar esta solução é necessário aplicar no eixo rotativo um íman, que uma vez colocado, variará um campo magnético. Com recurso a sensores magnéticos é possível detetar a frequência de rotação do contador e converte-la em consumos.



Figura 2.3: (a) Contador analógico alterado com tecnologias de *encoder* [12] (b) Contador analógico alterado com tecnologias de pulso magnético [13]

Aplicação destes métodos são inconvenientes, pois para cada contador é necessário aplicar uma abordagem diferente, e logo não universal. Por outro lado, os métodos baseados em sistemas de visão, não necessitam de alterações físicas ao contador analógico, e a sua aplicação é universal, pois depende do software desenvolvido. A figura 2.4, apresenta *smartphone* aplicado a um contador analógico.



Figura 2.4: Contador digital inteligente [14]

Portanto, a abordagem pode ser aplicada a vários contadores analógicos ou digitais. Esta abordagem, pelas utilidades que apresenta será explorada no âmbito desta dissertação.

2.4 Leitura em contadores analógicos através de sistemas de visão

Aswathy [15], apresenta um estudo sobre as vantagens dos sistemas de visão aplicados aos contadores. Este sistema, captura a imagem frontal do contador e envia-a através de sistemas de comunicação sem fios para um computador central. Nesse computador, a imagem é vista pelo utilizador, que regista o valor do contador. O sistema foi montado, com recurso

a um *smart phone*, que tem câmara integrada e ligação à *internet*. A figura 2.5, apresenta o esquema geral do sistema.

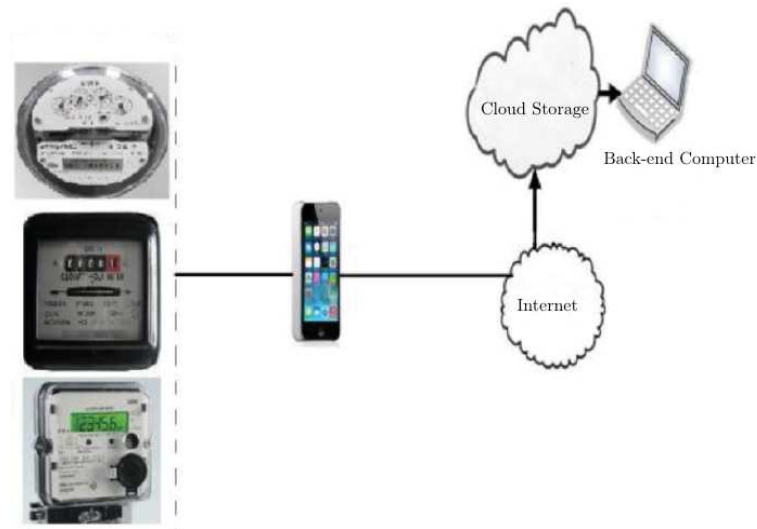


Figura 2.5: Sistema de transmissão dados através do uso de *smartphone* [15]

Este sistema tem a desvantagem de necessitar transmitir a imagem frontal do contador até ao Servidor (*Back-end*). Terminando a transferência, são aplicados tratamentos de imagem e obtidos os valores da contagem. Neste processo são aplicados tratamentos de imagem e algoritmos *Optical Character Recognition (OCR)*.

2.5 Algoritmos de processamento de imagem

Uma preocupação na aplicação dos sistemas de visão, consiste em determinar qual o nível de fiabilidade dos valores lidos.

Rodriguez [16], aborda o desenvolvimento de um algoritmo para reconhecimento de algarismos em contadores analógicos, denominado de *Hausdorff distance* (HD). Nos ensaios experimentais efetuados, obteve resultados de acerto na ordem dos 99,9%, e tempo máximo de reconhecimento de 31 milissegundos (ms).

Neste caso de estudo, o mais relevante é perceber de que modo o algoritmo de leitura está estruturado, e quais os pontos mais relevantes para a taxa de acerto elevadas.

2.6 Reconhecimento aplicado à imagem

Os sistemas de visão aplicados aos contadores dividem-se em três etapas principais. A "Captura de imagem", este processo consiste na aquisição da imagem frontal do contador pela câmara de vídeo.

O "Pré-processamento", onde são aplicados tratamentos à imagem, possibilitando a aplicação de algoritmos *OCR*.

Por último, o "Reconhecimento do número", que consiste na aplicação de algoritmos *OCR*, que permitem determinar qual o valor lido no contador. A figura 2.6 apresenta o processo descrito.

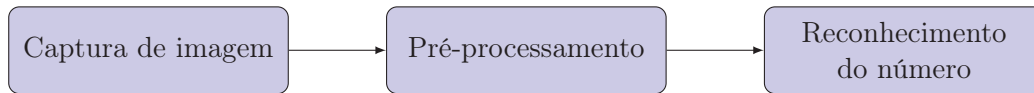


Figura 2.6: Principais processos para reconhecimento de imagem [16]

Na etapa, "Captura de imagem", é determinada a qualidade final da imagem frontal do contador. Esta etapa é relevante e pode influenciar a eficácia global do sistema.

A figura 2.7, representa de que modo o autor do artigo [16] posiciona a câmara perante o contador analógico, existindo substancialmente duas condições a ter em consideração nesta operação.

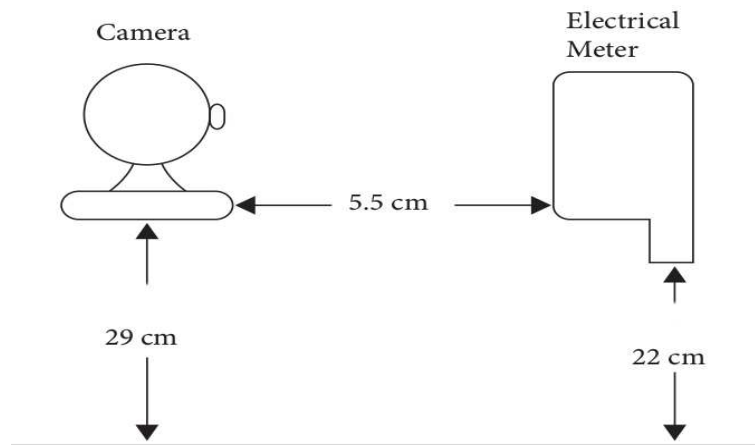


Figura 2.7: Distância e condições de captura da imagem [16]

A primeira condição refere-se à distância e ângulo relativamente ao contador, onde a câmara deve posicionar-se paralelamente ao ecrã frontal do contador, a uma distância que permita obter a imagem focada.

A segunda condição refere-se às condições de iluminação. Em ambientes escuros poderá ser necessário utilizar iluminação artificial e garantir que não haja reflexões na imagem obtida.

O segundo passo consiste no pré-processamento de imagem. Xie [4], apresenta um estudo que utiliza uma câmara *wireless*, onde captura o painel frontal dos contadores analógicos e envia essa informação para um servidor, onde é aplicado o pré-processamento de imagem e reconhecimento dos algarismos. A informação guardada permite a elaboração de gráficos dos consumos efetuados. O diagrama de blocos 2.8 apresenta o processo completo, desde a aquisição da imagem pela câmara, até à apresentação dos resultados das respetivas leituras.

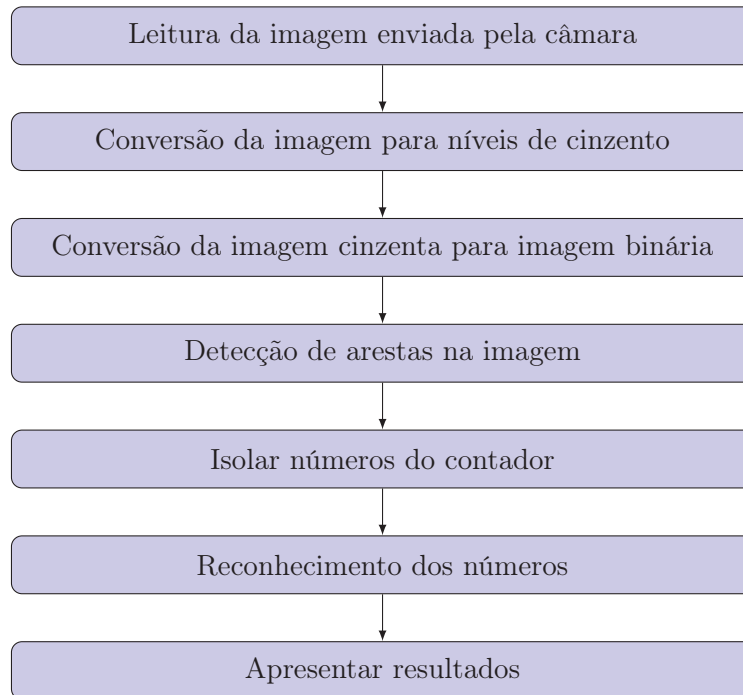


Figura 2.8: Processo geral de leitura [4]

O último passo consiste em reconhecer o número no contador, que é executado com base em algoritmos de *Optical Character Recognition (OCR)*. Este processo funciona com base na comparação de padrões (*patterns*), que são arranjos ou combinados de descritores (*features*).

Para reconhecer os algarismos do contador, é possível recorrer a dois algoritmos. O primeiro denominado *Tesseract*, foi desenvolvido pela empresa *Hewlett-Packard (HP)* entre as datas de 1985 e 1994, adquirido posteriormente pela empresa *Google*. O segundo *FineReader*, foi desenvolvido pela empresa *ABBYY* e tinha a finalidade de converter os documentos *PDFs* e fotografias digitais em arquivos editáveis.

2.7 Comparação entre *Tesseract* e *FineReader*

Heliński [17], apresenta um estudo comparativo entre o algoritmo *Tesseract* e o *FineReader*. Este tem o propósito de compreender qual dos algoritmos apresenta resultados com níveis de acerto mais elevados.

Os ensaios executados, recorreram a imagens "limpas" e "reais", ou seja, com e sem ruído. Em ambos, foram lidos 39 caracteres com o estilo de letra *antiqua* e *gothic*. As imagens utilizadas possuem a resolução de 800 * 600 pixels com fundo branco e letras pretas. A tabela 2.1 apresenta a percentagem de acerto em cada ensaio executado.

Tabela 2.1: Comparação entre os algoritmos *OCR FineReader* e *Tesseract* adaptada de [16]

Nome	Documento	Pagina	<i>FineReader</i> (%)	<i>Tesseract</i> (%)
Teste 1	<i>antiqua</i>	Limpa	89,97137	84,81774
Teste 2	<i>antiqua</i>	Original	83,08036	69,38797
Teste 3	<i>antiqua</i>	Limpa	89,68817	86,52127
Teste 4	<i>antiqua</i>	Original	69,39818	75,21368
Teste 5	<i>antiqua</i>	Limpa	90,89958	78,31711
Teste 6	<i>antiqua</i>	Original	80,14699	39,37743
Teste 7	<i>gothic</i>	Limpa	84,95764	85,94833
Teste 8	<i>gothic</i>	Original	82,37332	70,61988
Teste 9	<i>gothic</i>	Limpa	81,85919	87,26151
Teste 10	<i>gothic</i>	Original	73,24994	66,64538
Teste 11	<i>gothic</i>	Limpa	73,98262	80,64538
Teste 12	<i>gothic</i>	Original	52,79123	70,99617
Teste 13	<i>gothic</i>	Limpa	83,78564	91,71215
Teste 14	<i>gothic</i>	Original	80,47636	83,90822
Teste 15	<i>gothic</i>	Limpa	82,50946	86,93861
Teste 16	<i>gothic</i>	Original	79,22829	60,16857
Teste 17	<i>gothic</i>	Limpa	85,10663	86,12579
Teste 18	<i>gothic</i>	Original	84,00548	77,30411

A conclusão apresentada afirma que ambos os algoritmos não obtêm perpetuamente 100% de acertos. Todavia, é possível quantificar que o algoritmo *Tesseract* alcançou um acerto médio de 79,5% e o *FineReader* 82,4%. Portanto, ambos são equivalentes relativamente às taxas de acerto. O autor do estudo, afirma que é mais intuitivo utilizar o algoritmo *Tesseract*, razão pela qual será este utilizado nos ensaios práticos.

2.8 Pré-processamento e reconhecimento aplicado ao contador

Segundo Rodriguez [16], para que os algoritmos *OCR* reconheçam eficazmente os algoritmos, é necessário aplicar um pré-processamento de imagem e definir uma *region of interest* (*ROI*). Esta permite limitar a zona de leitura, resultando numa diminuição da quantidade de dados a serem processados. A figura 2.9, apresenta uma imagem frontal do contador sem qualquer tratamento, com pixels em níveis de cinzento, adquirida por uma câmara de vídeo e tem resolução de 1280 * 960 pixels.



Figura 2.9: Imagem original obtida pela câmara de vídeo

Após aquisição da imagem, é aplicado o tratamento. Este consiste em definir a *ROI*, onde as suas dimensões podem variar, porém, deve conter os dígitos do contador, que é a informação que se pretende ler. Assim, procede-se à aplicação da binarização (*Threshold*) da imagem. Neste processo os pixels cinzentos que variavam entre 0 até 1, ficam agora definidos entre 0 ou 1, consoante do valor nível para o *Threshold* aplicado.

Após aplicada a binarização à imagem, um pixel ou conjunto de pixels unidos entre si com valor 1, são estabelecidos como objetos. Os restantes que se tornam 0 (usualmente em superior quantidade) são estabelecidos como fundo da imagem. Na figura 2.10, os objetos "000 481" são posteriormente analisados pelos algoritmos *OCR*.



Figura 2.10: Pré-processamento da imagem frontal obtida pela câmara de vídeo

2.8.1 Leitura dos algarismos em movimento

Como referido anteriormente, os contadores analógicos, possuem uns anéis rotativos que são responsáveis por mostrar os consumos efetuados. Estes, quando estão no processo de rotação, podem tornar-se ilegíveis, pois, são apresentados de forma parcial. Este problema, deve ser contornado, pois, pode levar a erros no processo de leitura. A figura 2.11, apresenta a imagem frontal de um contador, num processo de transição.

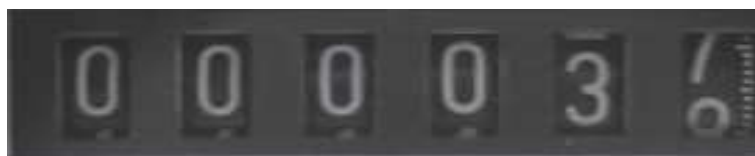


Figura 2.11: Contador analógico e algarismo em posição indefinida

Este obstáculo, pode ser ultrapassado, se for determinado uma percentagem de visibilidade para cada algarismo. A figura 2.12, apresenta um exemplo, onde os algarismos estão níveis de visibilidade de 83% e 17%. Todavia a 100%, estariam perfeitamente alinhados com a câmara.

Neste exemplo, a parte superior corresponde a 17% e a parte inferior a 83% de visibilidade.



Figura 2.12: Visibilidade dos algarismos na imagem pré-tratada

Aplicação deste método, permite definir um limite de visibilidade para cada algarismo. Este processo evitará que o algoritmo leia erradamente “partes de números”, que resultaria em dados indesejados. A figura 2.13 apresenta o fluxograma completo do algoritmo.

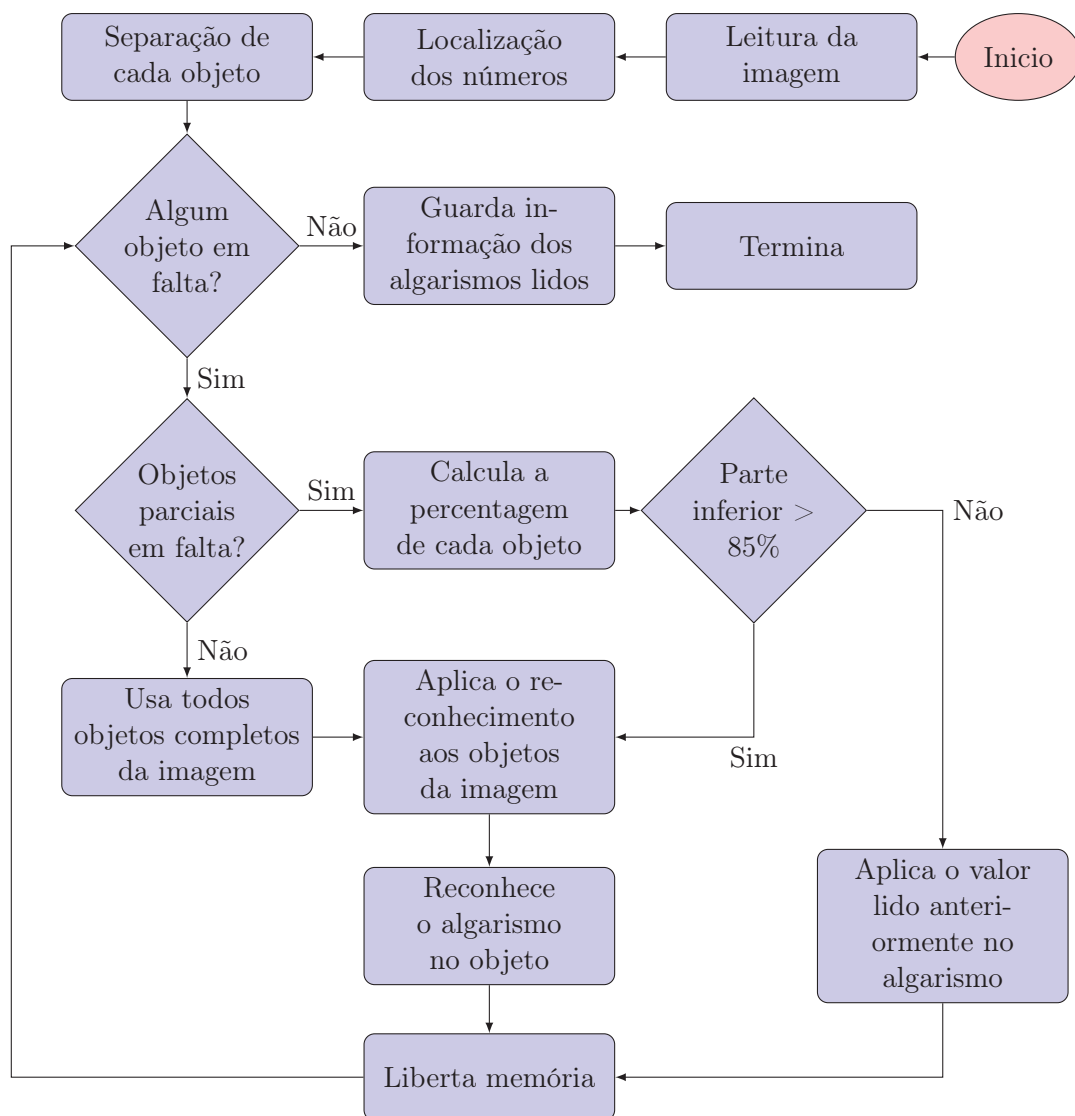


Figura 2.13: Fluxograma do algoritmo de leitura [16]

Rodriguez [16], aplica um método que permite validar ou descartar o valor final resultante

do processo de leitura. A validação é feita através da comparação do valor atual, com o lido previamente. Sendo descartado, se o atual for inferior ao anteriormente lido.

2.9 Soluções comerciais existentes

O propósito desta secção, é a análise de produtos comerciais, cujas características são comparáveis ao trabalho que se pretende desenvolver. Pela pesquisa feita, obtiveram-se diversas soluções telemétricas, havendo discrepâncias nos modos de funcionamento, alimentação do equipamento, modo de comunicação e o género de medição efetuada. Estas características foram as mais relevantes nesta pesquisa.

2.9.1 Equipamento *Metlink*

O equipamento *Metlink*, foi desenvolvido pela empresa *Teliko*. Este modelo permite fazer leituras de água, gás e eletricidade, e enviar os consumos efetuados através de tecnologias sem fios *3G* e *GSM*. A figura 2.14 ilustra a arquitetura do sistema.



Figura 2.14: Sistema *Metlink* [18]

Pela análise dos consumos, este possui ferramentas que permitem detetar falhas no fornecimento de energia, erros nas medições efetuadas e leituras fora dos padrões normais.

Os alertas são enviados para o utilizador através de *SMS* ou *email*. Estes avisos podem ser enviados para diversos equipamentos como *tablets*, telemóveis ou computadores.

Em cada mensagem enviada, é possível aceder aos dados sobre os consumos através de gráficos e tabelas.

2.9.2 Equipamento *RS2000*

O dispositivo *RS2000*, desenvolvido pela empresa *CAS*, permite fazer leituras de água e gás. A figura 2.15, mostra uma imagem do produto desenvolvido pela empresa *CAS*.

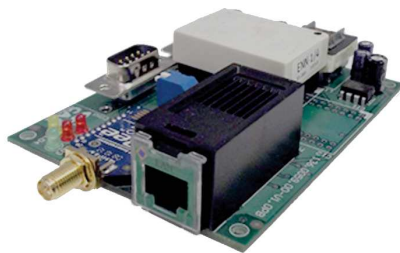


Figura 2.15: Equipamento *RS2000* [19]

Os dados lidos, são enviados para uma central, através de tecnologias de comunicação *GSM*, *GPRS* e *EDGE*. Onde posteriormente podem ser acedidos por diversos dispositivos como *tablets*, telemóveis ou computadores, com recurso a gráficos ou tabelas.

2.9.3 Equipamento *DD3G*

O produto mostrado na figura 2.16 é desenvolvido pela empresa *Duodigit*, havendo a possibilidade de monitorizar águas, gás e óleo.

Este equipamento tem suporte para duplo *SIM CARD*, que é uma vantagem, principalmente em zonas com sinal de comunicações fraco. É possível transmitir os dados via *3G*, *EDGE* e *GSM*, e vem equipado com as interfaces *RS232* e *usb 2.0*



Figura 2.16: Equipamento *DD3G* [20]

O sistema permite ler os dados automaticamente, que são posteriormente enviados a uma central, onde são monitorizados.

2.9.4 Equipamento *B280 prime*

O dispositivo *B280 prime*, apresentado na figura 2.17 foi desenvolvido pela empresa *Janz*. Este desenvolveu-se unicamente para fazer medições de eletricidade. Os dados podem ser enviados através do sistema de comunicação *GSM*, *GPRS* e *Ethernet*.



Figura 2.17: Equipamento *B280 Prime* [21]

Os consumos medidos, podem ser acedidos através de uma aplicação, que possui ferramentas onde é possível visualizar gráficos, diagramas, máximos e mínimos [15].

2.9.5 Resumo sobre os equipamentos analisados

Dos dispositivos examinados, meramente o *B280* não permite fazer contagem de água ou gás. Relativamente às comunicações todos os equipamentos optaram pela utilização de tecnologias *GSM*. Na Tabela 2.2, apresenta uma lista das principais características dos produtos.

Tabela 2.2: Resumo dos produtos do mercado comercial

Modelo	Contadores alvo	Comunicação
Metlink	Água, gás e eletricidade	3G, GSM
RS2000	Água e gás	GSM, GPRS, EDGE
DD3G	Água, gás e óleo	3G, EDGE, GSM, RS232 e USB 2.0
B280	Eletricidade	GSM, GPRS, Ethernet

É de salientar que nenhum dos equipamentos analisados permite reaproveitar os contadores analógicos já instalados e, portanto, é necessário fazer a sua substituição no local da instalação.

Esta característica é relevante, pois, é a base do trabalho que se pretende desenvolver. Relativamente aos preços dos equipamentos, estes variam entre 200 e 250 euros [22].

2.10 Rede de contadores ligados através da *Internet*

Tang [14], apresenta na figura 2.18 uma rede é constituída por contadores inteligentes e analógicos que foram reaproveitados através de novas tecnologias.

Neste caso de estudo foram utilizados *smartphones*, que capturam e enviam fotos pela *Internet* para um servidor principal. Uma vez recebidas as imagens, são aplicados tratamentos de imagem e extraídos os valores dos contadores. Em contrapartida, os contadores inteligentes, enviam a informação do consumo energético através da *internet*, que fica alocada diretamente numa base de dados.

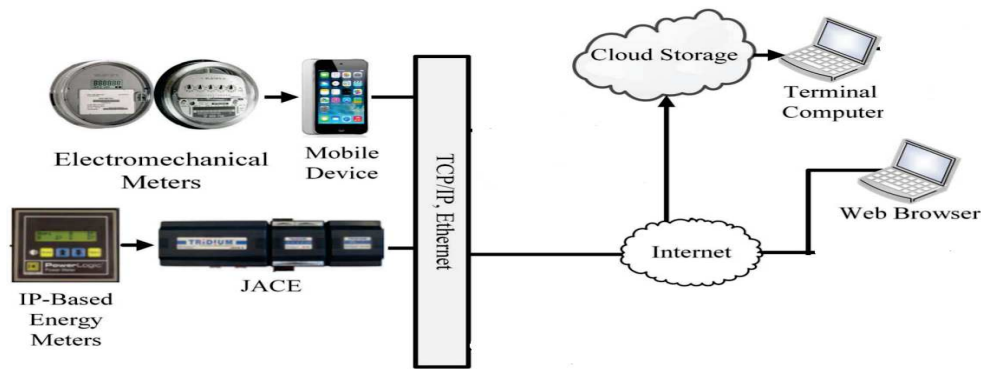


Figura 2.18: Rede de ligação adaptada de [14]

No final deste processo, os dados podem ser analisados com recurso a tabelas ou gráficos. Pela aplicação de *software* de análise de dados, é possível detetar possíveis fugas na rede de distribuição de água ou em habitações particulares e visualizar graficamente os respetivos consumos em cada um dos casos.

2.11 Detecção de fugas nas redes de distribuição de água

Hunaidi [23] refere que, na generalidade dos casos, os sistemas de distribuição de água têm perdas no transporte, que vão desde a estação de tratamento até aos consumidores. Tipicamente estes sistemas, em especial os mais antigos, a percentagem de perdas pode atingir os 50% .

A água desperdiçada pode ter causas como uso público para combate aos incêndios, fugas na rede de distribuição, etc.

As perdas de água provocadas pelas fugas, podem ocorrer em tubagens, junções, válvulas, etc. Estes fatores podem ocorrer devido à corrosão, material com defeito, instalação defeituosa, excessiva pressão de água dentro das canalizações, movimentos de terra, cargas excessivas sobre as tubagens e vibrações provocadas pelo trânsito, etc.

As fugas de água traduzem-se em prejuízos monetários e perdas de um recurso natural precioso limitado, podendo levar a um risco potencial na saúde da população.

A aplicação de auditorias, permite detetar abundantes quantidades de água desperdiçada. Este processo, pode ser aplicado ao sistema de distribuição, que requer detalhes sobre a quantidade de água que entra e sai no sistema, estes dados são fornecidos pelos medidores instalados nas redes.

Os contadores instalados na rede de abastecimento, usualmente podem ser constituídos por diversos medidores. A figura 2.19, permite observar que a diferença nos valores dos contadores do mesmo ramo é zero. Assim, idealmente o valor no Contador 4 é a soma do valor do Contador 5 com o Contador 6.

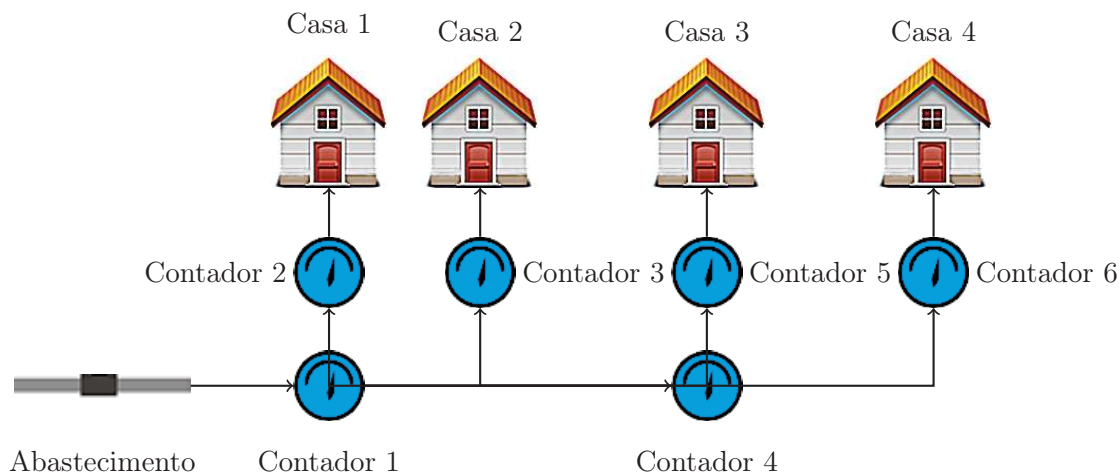


Figura 2.19: Conjunto de contadores aplicados à rede de abastecimento

O estudo preliminar da auditoria, pode ser executado com a seguinte informação e cálculos. Inicialmente, é necessário determinar a quantidade de água que entra no sistema (ES em m^3) de distribuição, dados que são tipicamente relativos a um ano de consumo. Posteriormente procede-se ao cálculo do consumo autorizado (CA em m^3). Assim, procede-se ao cálculo das perdas de água (PDA em m^3) através da equação 2.1.

$$PDA = ES - CA \quad (2.1)$$

- PDA = Perdas de água
- ES = Entrada no sistema
- CA = Consumo autorizado

Após o cálculo às perdas de água, procedem-se as perdas reais (PR em m^3), para tal é necessário estimar as perdas aparentes (PA em m^3). Estas incluem os consumos não autorizados, medições erradas dos contadores, erros de faturação e ajustamentos. Após este passo procede-se ao cálculo das perdas reais pela equação 2.2.

$$PR = PDA - PA \quad (2.2)$$

- PR = Perdas reais
- PA = Perdas aparentes

Após o cálculo das perdas reais, é possível calcular os custos dos prejuízos.

2.12 Identificação de fugas em habitações particulares

As fugas em habitações particulares, vulgarmente são menores que nas redes de distribuição. Estas podem ser detetadas através de alterações no consumo medido pelo contador à entrada da habitação. Para verificar a existência de uma possível fuga, é necessário fazer a monitorização do contador de água, que habitualmente está à entrada da habitação, como representado na figura 2.20.

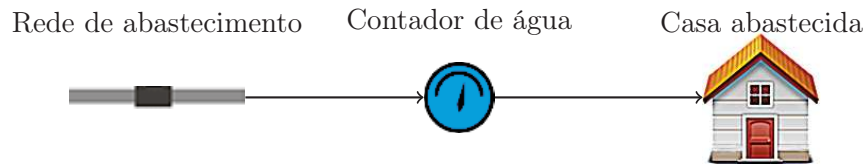


Figura 2.20: Rede de abastecimento de uma habitação

Inicialmente, deve-se garantir que todas as saídas de água estão devidamente fechadas e assegurar que não serão utilizadas por período mínimo, como, por exemplo, uma hora [24].

Antes de iniciar a experiência, regista-se o valor do contador. Após este passo, espera-se pelo tempo inicialmente estabelecido, e volta-se a efetuar uma leitura. A diferença entre o valor lido inicialmente e o último, corresponde à fuga no sistema.

2.13 Conclusão do estado de arte

Como referido previamente, é possível fazer reaproveitamento dos contadores analógicos e digitais de formas distintas, todavia, o método com sistemas de visão traz proveitos relativamente à aplicabilidade do sistema. Pois, permite reaproveitar os contadores analógicos e digitais, não necessitando de alterações significativas ao contador que se pretende aplicar novas tecnologias.

Dos sistemas analisados que recorreram a sistemas de visão, nenhum executa o reconhecimento de imagem no local onde se encontra instalado. Este método, necessita transmitir elevadas quantidades de dados, o que torna impraticável em ambientes industriais ou zonas remotas, onde os sinais de *internet* sejam fracos ou instáveis.

A aplicação de sistemas de visão verificou taxas de acerto na ordem dos 99,9%, através do uso de algoritmos *OCR* e com recurso a técnicas validação dos dados.

Ao nível industrial, os equipamentos analisados, nenhum permite o reaproveitamento dos contadores antigos instalados na rede, logo é necessário substituir na totalidade dos contadores antigos. Este processo implica custos que variam entre 200 e 250 euros, na aplicação do contador inteligente.

Relativamente à verificação de fugas, é possível detetar a existência das mesmas em habitações particulares, pela análise de consumos, e nas redes de distribuição, pela diferença de consumos em vários contadores instalados.

Assim, é viável o desenvolvimento de um equipamento apto para ler vários contadores. Que tenha capacidade de transmitir os dados lidos para um servidor, gerir consumos e alarmes sobre as possíveis fugas na rede de abastecimento, a um custo inferior aos praticados pelo mercado comercial.

Capítulo 3

Solução de *hardware* proposto

Inicialmente, este projeto idealizou-se para ser desenvolvido numa arquitetura baseada em microcontroladores. No entanto, após uma longa pesquisa sobre adaptação das bibliotecas de sistemas de visão a microcontroladores, relevou-se complexa a sua aplicabilidade. Todavia, a mesma pesquisa relevou que, as mesmas bibliotecas aplicadas ao sistema operativo *Linux*, não são tão complexas. Assim, optou-se por desenvolver o *software* do equipamento em linguagem *C* e *C++* utilizando o sistema operativo *Linux*.

3.1 Requisitos para desenvolvimento do produto

O equipamento, deve ter capacidade de ler o valor medido no contador analógico, que se encontra instalado nas redes de abastecimento, e ter sistemas que permitam detetar possíveis fugas nas instalações. O leitor após executar um determinado número de leituras ao contador, num período de tempo pré-estabelecido, deve envia-las para o servidor. Neste, deve haver uma aplicação, responsável por receber os valores medidos pelos diversos leitores instalados e proceder à análise para deteção de possíveis fugas nas redes de abastecimento.

Pela pesquisa efetuada, o equipamento deve ser capacitado para efetuar leituras aos contadores analógicos, pelo uso de sistemas de visão. Consequentemente, é conveniente aplicar bibliotecas de sistemas de visão, e algoritmos *OCR* que permitam ler o valor no contador analógico.

Para que a taxa de acerto aumente na aplicação dos algoritmos *OCR*, deve ser garantida a qualidade da imagem da câmara de vídeo. Desta forma, é necessário a utilização de iluminação artificial, com uso de níveis de intensidade e dispersão propícia, impedindo possíveis reflexões. No posicionamento da câmara de vídeo, deve-se garantir uma fixação correta, em distância e alinhamento.

Uma vez que se pretende que haja a possibilidade de transmissão os consumos, é necessário aplicar sistemas de comunicação. A utilização de tecnologias, *WIFI*, *GSM* e *GPRS* permitem a transmissão dos dados para um servidor principal, que é responsável por analisar os consumos medidos pelo contador.

O módulo de processamento, deve possuir portas *USB* ou *General Purpose Input/Output (GPIO)*, que são fundamentais para ligar a câmara e dispositivos de comunicação sem fios.

Por último, a unidade de processamento (*CPU*) e memória *RAM* são relevantes, mas dependem do *software* desenvolvido.

3.2 Estrutura da unidade de leitura

A figura 3.1, representa o dispositivo, que é constituído pela unidade de processamento, câmara de vídeo, iluminação artificial difusa e módulo de *GSM*.

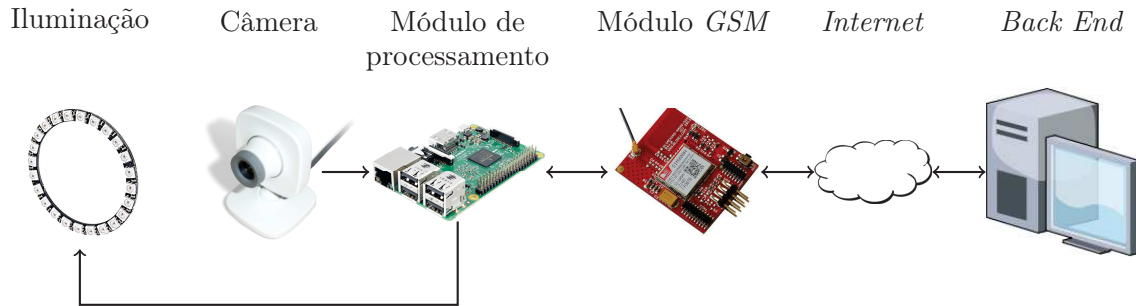


Figura 3.1: Estrutura do modelo desenvolvido

O módulo de processamento é responsável por controlar a iluminação artificial, de modo a garantir a qualidade das imagens. Deve igualmente obter e guardar a imagem na memória temporariamente, processá-la, guardar o valor do contador e enviá-lo se necessário para o servidor. O envio de dados pode ser executado através de *GSM*, *GPRS* ou *WIFI*.

3.2.1 Unidade de processamento

Esta secção pretende analisar as placas de desenvolvimento existentes no mercado comercial. Esta análise, consideraram-se requisitos de *Hardware* e *Software*, suportados por cada equipamento. O nível de comunidade, refere-se à quantidade de pessoas que desenvolvem *software*, aplicado a esses equipamentos.

Raspberry Pi

A empresa *Raspberry Pi*, possui diversas placas de desenvolvimento, no entanto, selecionou-se dois modelos. A placa *Raspberry Pi W*, de baixo custo e *Raspberry Pi 3* de valor mais elevado. Ambas as suas características são apresentadas na tabela 3.1, onde é possível analisar que são equivalentes e possuindo discrepâncias sobretudo ao nível de *CPU*, memória *RAM* e preço.

Tabela 3.1: Comparação dos equipamentos da empresa *Raspberry Pi*

Equipamento	<i>Raspberry Pi W</i>	<i>Raspberry Pi 3</i>
S.O Sportado	<i>Raspbian, Ubuntu</i>	<i>Mate, Windows 10 iot core</i>
Comunidade	Alta	Alta
<i>CPU</i>	<i>ARM11 a 1 GHz</i>	<i>4*ARM Cortex-A53,1.2 GHz</i>
<i>RAM</i>	512 MB	1 GB
<i>GPIO</i>	<i>40-pin GPIO</i>	<i>40-pin GPIO</i>
<i>Wireless</i>	Sim	Sim
<i>Bluetooth</i>	Sim	Sim
<i>Output</i>	<i>Micro USB</i>	<i>4*USB 2.0,Ethernet,Câmara serial interface</i>
<i>Storage</i>	<i>MicroSD</i>	<i>MicroSD</i>
<i>Video&Audio</i>	<i>HDMI</i>	<i>HDMI</i>
Preço	5 euros	39,9 euros

Arduino

À semelhança da empresa anterior, o *Arduino* também possui diversas placas de desenvolvimento. Foram selecionadas a placa *Arduino Uno R3* e *Arduino MKR Zero*, ambas com as características apresentadas na tabela 3.2. É possível verificar que são discrepantes, no entanto, limitadas principalmente na comunicação sem fios, memória *RAM* e velocidade de processamento (*CPU*).

Tabela 3.2: Comparação dos equipamentos da empresa *Arduino*

Equipamento	<i>Arduino Uno R3</i>	<i>Arduino MKR Zero</i>
S.O Sportado	<i>Arduino Software (IDE)</i>	
Comunidade	Alta	Alta
<i>CPU</i>	16 MHz	48 MHz
<i>RAM</i>	2 KB	32 KB
<i>GPIO</i>	14-pin	22-pin
<i>Wireless</i>	Não	Não
<i>Bluetooth</i>	Não	Não
<i>Storage</i>	32 KB	<i>MicroSD</i>
<i>Video&Audio</i>	Não	Não
Preço	7.53 euros	18.6 euros

Orange

As placas da empresa *Orange* possuem algumas semelhanças relativamente às características apresentadas pela *Raspberry Pi*. Foram igualmente selecionados duas placas, a *Orange Pi 2G-LoT* e a *Orange Pi One*. As suas características são apresentadas na tabela 3.3. Os equipamentos são equivalentes, sendo a placa *Orange Pi 2G-LoT* ímpar a apresentar uma solução com comunicação *2G GSM/GPRS* embutida, relativamente às placas concorrentes.

Tabela 3.3: Comparação dos equipamentos da empresa *Orange*

Equipamento	<i>Orange Pi 2G-IoT</i>	<i>Orange Pi One</i>
S.O Sportado	<i>Android 4.4, Ubuntu, Debian, Raspbian image</i>	
Comunidade	Baixa	Baixa
<i>CPU</i>	1 GHz	800 MHz
<i>RAM</i>	256 MB	512 MB
<i>GPIO</i>	<i>40-pin GPIO</i>	<i>40-pin GPIO</i>
<i>Wireless</i>	Sim	Não
<i>Bluetooth</i>	Sim	Não
<i>Output</i>	<i>USB, 2G GSM/GPRS</i>	<i>Ethernet, 3*USB, USB OTG</i>
<i>Storage</i>	<i>MicroSD</i>	<i>MicroSD</i>
<i>Video&Audio</i>	Suporte para <i>LCD</i>	<i>HDMI</i>
Preço	10 euros	27.31 euros

Asus e MediaTek

As empresas *Asus* e *MediaTek* apresentam duas placas deveras recentes. A *Asus Tinker* e *MediaTek X20*, é visível na tabela 3.4 ambas são similares ao nível de *Hardware*. Todavia, discrepantes ao nível de *Software* suportado pelo sistema, onde a placa *MediaTek X20* suporta exclusivamente *Android 6.0*.

Tabela 3.4: Comparação dos equipamentos *Asus* e *MediaTek*

Equipamento	<i>Asus Tinker</i>	<i>MediaTek X20</i>
S.O Sportado	<i>Android, Ubuntu, Debian</i>	<i>Android 6.0</i>
Comunidade	Baixa	Baixa
<i>CPU</i>	1.8 GHz	2.3 GHz
<i>RAM</i>	2 GB	2 GB
<i>GPIO</i>	<i>40-pin GPIO</i>	<i>40-pin GPIO</i>
<i>Wireless</i>	Sim	Sim
<i>Bluetooth</i>	Sim	Não
<i>Output</i>	<i>4*USB, Ethernet</i>	<i>2*USB</i>
<i>Storage</i>	<i>MicroSD</i>	<i>MicroSD</i>
<i>Video&Audio</i>	<i>HDMI</i>	<i>HDMI</i>
Preço(euros)	75 euros	167 euros

3.2.2 Resumo sobre placas pesquisadas

Pela pesquisa feita, as placas *Orange Pi 2G-IoT* e *Raspberry Pi W*, apresentam as melhores características para desenvolvimento do projeto, relativamente à qualidade e preço. A *Orange Pi 2G-IoT*, em particular, tem a característica de suportar comunicações *2G GSM/GPRS*. Esta funcionalidade é útil, em especial para os contadores analógicos localizados em zonas remotas, sem redes comunicações *WIFI*.

As placas *Raspberry Pi 3*, *Orange Pi One* e *Aus Tinker*, possuem um custo mais elevado. No entanto, apresentam caraterísticas com capacidade relevantes para o projeto. O facto de suportar o sistema operativo *Ubuntu*, permite igualmente ter acesso à variedade de bibliotecas

para desenvolvimento de aplicações.

Por último, as placas *MediaTek X20*, *Arduino Uno R3* e *Arduino MKR Zero*, têm as características menos apelativas deste grupo. A *MediaTek X20*, independentemente de possuir características interessantes, o preço é elevado e só possui capacidade para suportar o sistema operativo *Android 6.0*, que comparativamente aos seus concorrentes, é limitativo. Outro ponto menos favorável é ter uma comunidade de desenvolvedores em menor quantidade.

As placas da empresa *Arduino*, são ambas limitativas no sistema operacional suportado e as características de velocidade de processamento, memória *RAM*, são inferiores às suas concorrentes.

3.2.3 Módulo de comunicação

O módulo de comunicação, pode não ser utilizado se na placa de desenvolvimento já vier embutida como acontece com a placa *Orange Pi 2G-LoT*, no entanto será apresentada uma placa para aplicação prática.

O equipamento *SIM800H*, é compatível com as placas de desenvolvimento *Arduino*, *Raspberry Pi*. A tabela 3.5 apresenta as principais características.

Tabela 3.5: Principais especificações do equipamento *SIM800H*

Equipamento	Módulo <i>SIM800H</i>
Alimentação	3.4 V a 4.4 V (2 A corrente de pico)
<i>Quad-Band</i>	850/900/1800/1900 Mhz
Ganho da antena	3 dBi
Temperatura de funcionamento	-40 °C a +85 °C
Dimensões (C*L)	15,8*17,8*2,4 mm
Peso	1,33 g
Preço	11,32 euros

O módulo apresentado permite fazer ligações de voz, enviar *SMS* e transferir dados via *GPRS* e *GSM*, ideal para locais remotos. Este possui *slot* para cartão *SIM* de tamanho convencional, pinos para conexão de auscultadores de ouvido e microfone. Para fazer a ligação às placas de desenvolvimento, são utilizados os pinos *I/O*.

3.2.4 Câmara de vídeo

A câmara de vídeo, tem a finalidade capturar a imagem frontal do contador analógico. Portanto, tem de cumprir com requisitos mínimos, como a resolução mínima e distância focal, que depende fortemente do encapsulamento utilizado. O encapsulamento tem a finalidade de bloquear interferências luminosas, e fixar a câmara e a iluminação artificial ao contador analógico.

A figura 3.2, apresenta a câmara *Microsoft* modelo *Xbox 360 Live Vision*, utilizada nos ensaios práticos.



Figura 3.2: *Xbox Live Vision* [25]

Na escolha da câmara de vídeo, um requisito relevante consiste no controlo sobre os parâmetros internos, como, o ganho para luminosidade, ou a distância focal. Estes parâmetros, alteram automaticamente e podem ocorrer circunstâncias que comprometa a leitura final. Na tabela 3.6, estão apresentadas as características da câmara utilizada no protótipo.

Tabela 3.6: Principais especificações da câmara *Microsoft Xbox 360 Live Vision*

Equipamento	<i>Microsoft Xbox 360 Live Vision</i>
Tipo ligação	<i>USB</i>
Resolução fotográfica	1280*960 pixeis
Resolução vídeo	800*600 pixeis a 30 fps
Focagem	Manual
Preço	32 euros

Este equipamento utiliza a porta *USB* e tem ajuste do foco manualmente. Esta última funcionalidade é relevante, pois, proporciona o foco da imagem a curtas distâncias, sem que seja alterada automaticamente.

3.2.5 Iluminação artificial

A iluminação artificial necessita igualmente de cumprir requisitos, para não comprometer a qualidade da imagem final. A figura 3.3, apresenta um exemplar de um dispositivo para iluminação que pode ser aplicado ao equipamento.



Figura 3.3: Conjunto de *LEDs* [24]

Para que a qualidade da imagem seja a mais elevada possível, é necessário garantir o mí-

nimo de reflexões provocadas pelo sistema de iluminação na câmara de vídeo. Estas reflexões podem levar à saturação de vários pontos na imagem final comprometendo o resultado.

O uso de iluminação à base de *LEDs*, têm benefícios, pois, não emitem calor, têm um consumo energético baixo e para o controlo da intensidade luminosa, é possível variar o brilho através de variação da corrente de alimentação.

3.2.6 Resultado da montagem do *Hardware*

Na montagem do *hardware* para os ensaios práticos, recorreu-se a um contador analógico de eletricidade. Neste aplicou-se o sistema de leitura constituído pelo encapsulamento, que fixa a câmara de vídeo e a iluminação artificial *LED*.

O contador, foi alterado mecanicamente para a execução dos ensaios. Assim é possível ligar, desligar a contagem recorrendo um interruptor e alterar a sua velocidade de rotação do painel, com recurso a potenciômetro.

A figura 3.4, apresenta uma foto do resultado prático na montagem dos componentes, nela é visível o encapsulamento e a câmara em frente ao contador analógico.

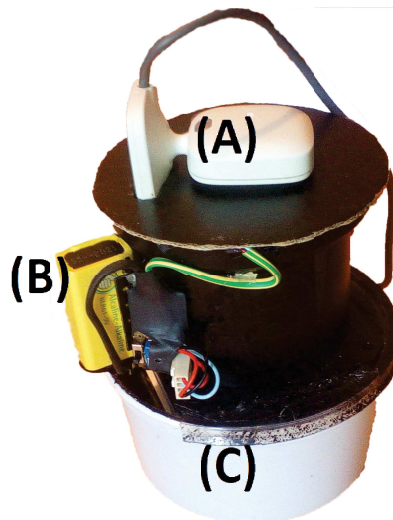


Figura 3.4: (A) Câmara de vídeo (B) Pilha de alimentação para a iluminação artificial aco-
plada ao encapsulamento (C) Base onde se encontra o motor para movimentar o contador

O encapsulamento é dividido em duas partes, a parte superior apresentado na figura 3.5, ponto (a) (que contém a iluminação e câmara) e inferior apresentada na figura 3.5, ponto (b) (está fixado no contador analógico). Ambas as partes quando encaixadas, permitem ajustar a distância da câmara ao contador, facilitando a focagem se necessário.

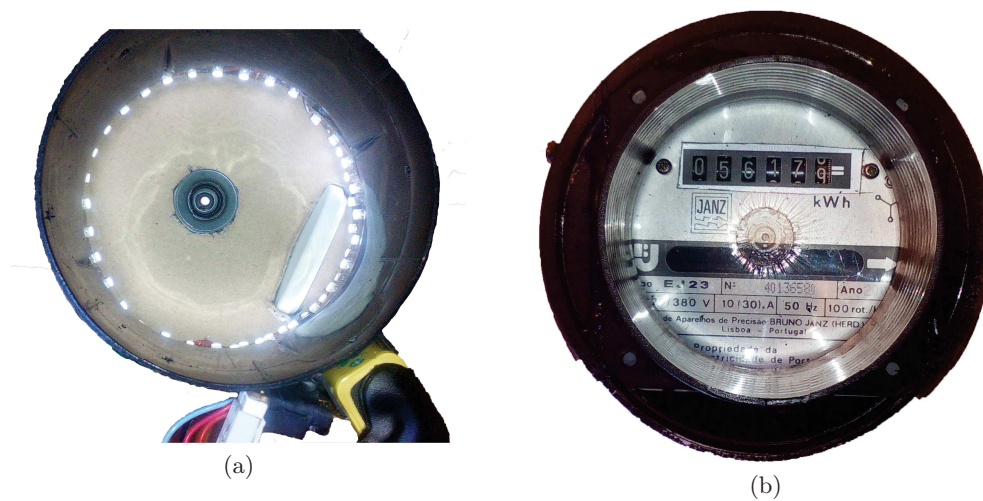


Figura 3.5: (a) Parte superior, com iluminação e câmara de vídeo (b) Parte inferior, com o contador

O encapsulamento, é composto pela iluminação artificial com fita de *LEDs*, colocada de forma circular e não direcionada para o painel frontal do contador. O sistema de iluminação é alimentado através de uma pilha de 9V.

Capítulo 4

Solução de software proposto

O *software*, como referido no capítulo 3, é desenvolvido no sistema operativo *Linux* numa distribuição *Ubuntu*, em linguagem *C* e *C++*. Paralelamente utilizou-se a ferramenta *Glade Interface Designer* (*GIP*), para desenvolver a interface com o utilizador. A aplicação é dividida em dois grupos. O primeiro grupo refere-se ao servidor (*Back-end*), que tem a finalidade de receber os dados enviados pelos diversos leitores aplicados aos contadores e detetar possíveis fugas nos dados recebidos pela diferença dos valores nas redes de distribuição. O segundo refere-se ao leitor instalado nos contadores, que tem a finalidade de proceder à aquisição e tratamento de imagem, reconhecimento de algarismos, guardar, validar, analisar e enviar os dados.

4.1 Requisitos do software

A informação neste sistema propaga-se desde a leitura efetuada pelo leitor aplicado ao contador analógico, que é enviada periodicamente pela *Internet*, utilizando serviços de comunicação *WIFI*, *GPRS* ou *GSM* até ao servidor.

A figura 4.1, apresenta esquematicamente a ligação entre os leitores aplicados ao contador e o servidor.

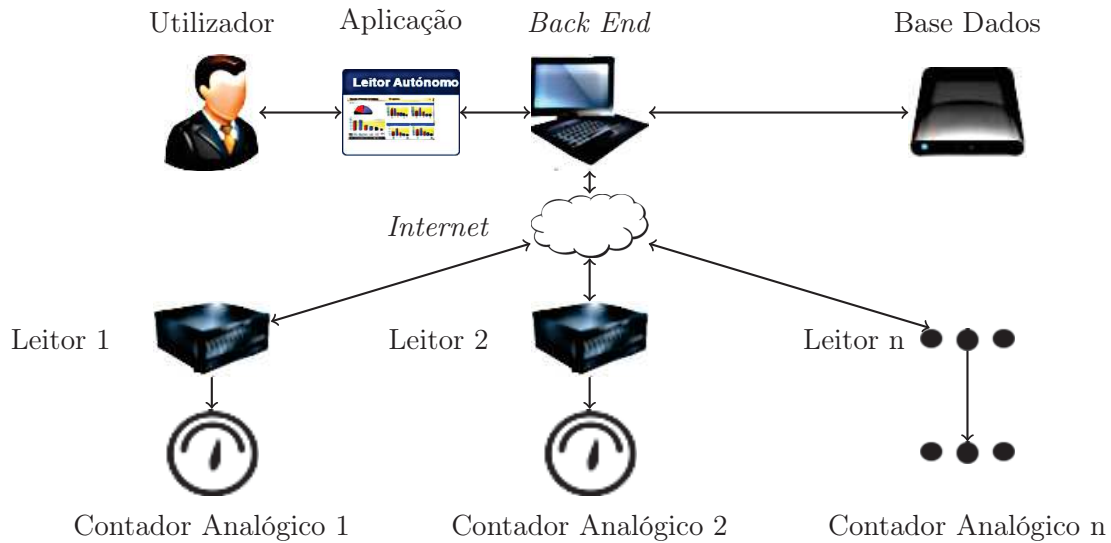


Figura 4.1: Esquema completo do sistema

O sistema é responsável por guardar dados recebidos, para que posteriormente possam ser acedidos pelo utilizador.

4.2 Introdução ao Back End

A aplicação desenvolvida para execução no servidor possui uma interface gráfica que permite ao utilizador aceder a diversas ferramentas. As leituras são enviadas pelos leitores, e armazenadas numa base de dados. Os dados recebidos, podem conter alerta de fuga em habitações particulares, resultante da análise dos consumos na habitação. As leituras enviadas pelos leitores instalados nas redes de abastecimentos podem igualmente gerar alarmes, através da diferença nos consumos efetuados pelos CA.

4.3 Principais funções do Back End

Esta secção tem a finalidade de apresentar as ferramentas que se pretende desenvolver no servidor, que permitem gerir dispositivos, consultar consumos, gerar alertas de fugas e fazer configurações.

A funcionalidade "Adicionar dispositivos" apresentada na tabela 4.1, tem a função registar dados sobre os novos leitores instalados em rede. Para efetuar o registo dos contadores, é necessário inserir dados sobre o nome do proprietário do leitor, localização *GPS*, seleção do tipo de leitura e período de envio de dados.

Tabela 4.1: Funcionalidade adicionar dispositivos

Nome da função	Adicionar dispositivos
Campo 1	Inserir o nome do proprietário
Campo 2	Inserir a localização do leitor (<i>GPS</i>)
Campo 3	Selecionar tipo de leitura (água, gás ou eletricidade)
Campo 4	Inserir <i>IP</i> da ligação (ex:192:168:001:001)

A segunda funcionalidade tem a finalidade de "consultar os dados" inseridos de cada leitor instalado, na qual as funcionalidades estão apresentadas na tabela 4.2.

Tabela 4.2: Funcionalidade consulta de dados

Nome da função	Consulta de dados
Campo 1	Inserir o número de registo do leitor que se pretende consultar
Campo 2	Apresenta nome do leitor
Campo 3	Apresenta localização
Campo 4	Apresenta o período registado
Campo 5	Apresenta ligação <i>IP</i>
Campo 6	Apresenta valores máximos, médios e mínimos dos dados recebidos
Campo 7	Apresenta total consumido até à data

Por último, a funcionalidade "Alerta de fugas nos contadores", apresentada na tabela 4.3, tem a finalidade de alertar o utilizador sobre os dois modelos de alarmes. O primeiro refere-se à constatação de fugas na rede privada, o segundo à deteção de fugas na rede pública. Sempre que um alarme é gerado, a aplicação deve abrir uma janela permitindo alertar o utilizador e apresentar os seguintes dados:

Tabela 4.3: Funcionalidade alerta de fugas

Nome da função	Alerta de fugas
Campo 1	Apresenta o número do leitor registado
Campo 2	Apresenta data da ocorrência do alarme
Campo 3	Apresenta o tipo de alarme ocorrido
Campo 4	Apresenta o cálculo de perdas estimadas até à data do alarme

4.4 Introdução ao *software* aplicado ao CA

O funcionamento dos leitores, divide-se em duas partes. A primeira refere-se ao tratamento de imagem, verificação de valores adquiridos, armazenamento de dados e envio destes para o servidor. A segunda, relaciona-se com a configuração para que o dispositivo funcione em modo autónomo. O fluxograma 4.2 apresenta o esquema completo para aplicação do sistema de leitura no contador analógico.

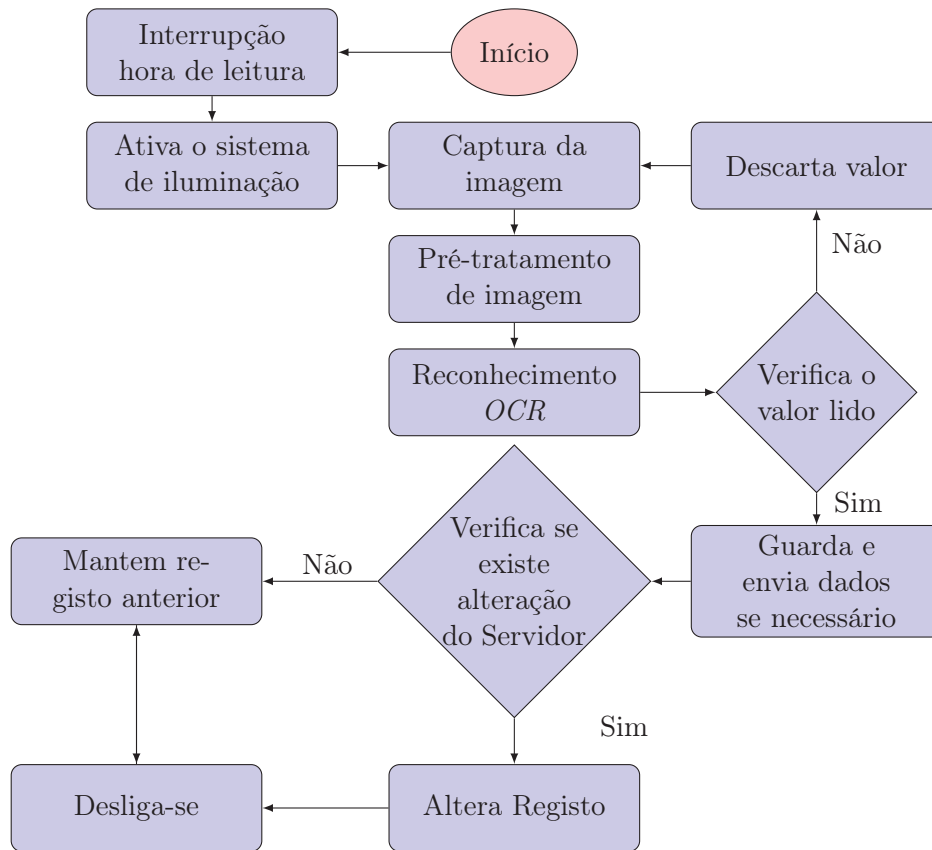


Figura 4.2: Fluxograma do sistema de leitura e envio de dados para o servidor

Na verificação do valor lido, é possível definir um número de tentativas, caso não seja possível fazer a leitura, então o agenda uma nova data e desliga-se.

4.5 Configuração da aplicação para leitura autónoma

Para que o sistema trabalhe de forma autónoma, deve-se proceder a configurações iniciais. Inicialmente, é necessário indicar o número de dígitos e o valor atual apresentado no painel do contador analógico. Ambos os dados são relevantes para controlo das leituras efetuadas pelo algoritmo desenvolvido. É igualmente necessário indicar o número de leituras executadas por mês, para que o sistema autonomamente as execute e envie os resultados para o servidor. Por último, é necessário definir a *Region of Interest ROI*, o nível de *Threshold* e *BlobsCleaner*, no qual a explicação é feita nos tópicos seguintes.

4.5.1 Seleção da *Region of Interest*

Como referido na subsecção 2.8, a *ROI* tem propósito de limitar a informação lida pelo sistema. Para tal, é necessário definir a zona de leitura aplicada à imagem do contador analógico. A figura 4.3, apresenta a imagem frontal do contador utilizado nos ensaios práticos, para exemplificação do processo de configuração.



Figura 4.3: Imagem do contador sem seleção de área para leitura

Para definir a *ROI*, selecionam-se dois pontos no ecrã, o ponto superior esquerdo e inferior direito da região que se pretende ler. A figura 4.4, apresenta a região selecionada para a leitura dos números do contador analógico.



Figura 4.4: Imagem do contador com seleção de área para leitura

Após selecionada a *ROI* de leitura, assim é aplicado o corte que está representado na figura 4.5.

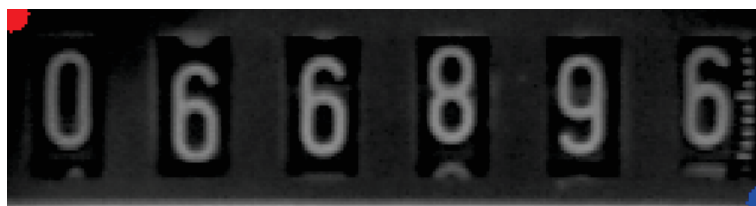


Figura 4.5: Imagem que irá ser sujeita aos tratamentos de imagem

Todos os tratamentos de imagem e leituras efetuadas são feitas exclusivamente na região determinada pela *ROI*, sendo logicamente a restante informação presente ignorada pelo sistema de leitura.

4.5.2 Binarização da imagem

No processo de binarização da imagem, foram utilizados dois métodos. O primeiro consiste na aplicação do *Threshold*, onde é necessário definir manualmente o nível aplicado à imagem, que varia de 0 a 255 (por ser uma imagem a níveis de cinzentos). A figura 4.6, apresenta figuras com nível insuficiente (a) e (b) com nível excessivo.



Figura 4.6: (a) *Threshold* a nível 46 (b) *Threshold* a nível 117

Neste processo, fica ao encargo do utilizador a escolha de um nível satisfatório, que permita alcançar resultados favoráveis. A figura 4.7, apresenta o nível ideal de *Threshold* a aplicar.



Figura 4.7: *Threshold* a nível 85

O segundo consiste na aplicação da binarização de forma automática, através da utilização do *Threshold* adaptativo. Assim o utilizador não tem qualquer impacto no valor escolhido, contrariamente ao modo manual fica ao encargo do utilizador a escolha de um valor favorável, o que poderá comprometer o resultado pretendido.

4.5.3 Aplicação do *BlobsCleaner*

A função *BlobsCleaner* é aplicada à imagem previamente binarizada e tem a finalidade de separar os números do ruído existente resultante do processo anterior. Esta função permite eliminar objetos pelo número de pixels que estes contêm na sua constituição, e formar uma nova imagem. Para tal, o utilizador seleciona manualmente o número de pixels por área que pretende eliminar. A figura 4.8, no ponto (a), apresenta a imagem original que contém os objetos. A imagem (b), apresenta os objetos que contêm mais de 2000 pixels, onde os demais foram eliminados.



Figura 4.8: (a) *BlobsCleaner* a nível 0 (zero) (b) *BlobsCleaner* a nível 2000

A figura 4.9, apresenta o nível a aplicar à imagem que deve garantir que os números não sejam eliminados.



Figura 4.9: *BlobsCleaner* aplicado a nível 845

Para compreender o funcionamento da função, a figura 4.10, apresenta outro exemplo semelhante que resultou da aplicação da binarização automática. A figura contém diversos objetos que não fazem parte dos números que se pretende ler, logo devem ser eliminados. Neste caso é necessário que os objetos "050 030", fiquem isolados dos restantes que se encontram na imagem.



Figura 4.10: Imagem do contador com aplicação do *Threshold*

Após aplicada a função *BlobsCleaner*, é criada uma imagem, mediante a área em pixels escolhida, que está representada na figura 4.11.



Figura 4.11: Imagem vinda da aplicação do *BlobsCleaner*

A imagem final é obtida pela subtração da imagem da figura 4.10, com a figura 4.11, resultando numa imagem limpa, com os algarismos devidamente isolados dos restantes objetos, como apresentada na figura 4.12.



Figura 4.12: Imagem final após aplicação do *BlobsCleaner*

O propósito deste processo é conseguir que os algarismos fiquem isolados e que não existam outros objetos, para além dos que se pretende ler na imagem, e que possam interferir com o processamento. Todavia, este processo nem sempre é possível obter uma imagem completamente "limpa", portanto foram desenvolvidos mecanismos que permitem lidar com essas circunstâncias, explicado na secção 4.6.

4.6 Verificação da posição dos algarismos no contador

Na subsecção 2.8 referiu-se que os algarismos rotativos do contador, poderão não ser legíveis aquando do processo de transição. De forma a contornar este problema, foi desenvolvido

um algoritmo que permite detetar essas transições no contador analógico. Esta secção tem a finalidade mostrar os erros que podem ocorrer na leitura no momento da transição dos algarismos no contador e como resolve-los. Para fins ilustrativos, foram seleccionadas duas imagens representadas na figura 4.13, onde os mostradores de ambas estão precisamente em transição.



Figura 4.13: Contadores (a) e (b) em transição

Após a aplicação da configuração inicial, ambas as imagens resultaram com dígitos "deformados", apresentados na figura 4.14.



Figura 4.14: Aplicado tratamento as figuras no exemplo (a) e (b)

Neste caso, os resultados da aplicação do algoritmo *OCR*, estão apresentados na figura 4.15.

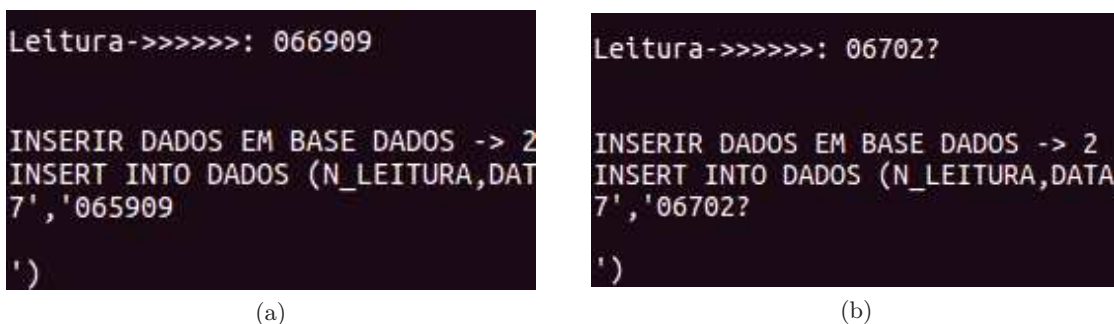


Figura 4.15: Resultados obtidos das leituras do exemplo (a) e (b)

Pelas imagens é visível que no primeiro caso o resultado de "066 909", onde o algarismo "9", não corresponde à leitura real do sistema. Em contrapartida, no segundo exemplo resultou o

valor de "06702?", do qual último algarismo corresponde erradamente ao símbolo "?".

Neste exemplo, ambos os resultados foram errados, logo podem também causar erros nas leituras finais do contador. Assim o momento de transição é um estado do contador que tem ampla importância, portanto é fundamental diminuir a probabilidade de ocorrer erros de leitura nestes casos.

Para resolver este inconveniente, desenvolveu-se um algoritmo que permite calcular a percentagem de visibilidade de cada um dos algarismos do contador, e fazer a respetiva leitura somente quando a percentagem estiver acima de um determinado limite.

O algoritmo, consiste na definição da altura e largura padrão dos objetos na imagem, através da aplicação da mediana dos números.

A ferramenta *BondingBox*, permite conseguir características dos objetos selecionados como, *centroid*, área, orientação, altura, largura, etc, estes parâmetros podem ser adquiridos em pixels ou milímetros.

Após obter a altura e largura de todos os objetos, são guardados os valores para que seja aplicado a mediana e determinar a largura e altura padrão a utilizar no algoritmo. A figura 4.16, apresenta a *BondingBox* aplicada aos objetos da imagem.



Figura 4.16: Modo de deteção no exemplo (a) e (b)

Sabendo a altura e largura padrão, é possível calcular a percentagem de visibilidade para cada um dos algarismos.

Pela percentagem mínima definida, os objetos da imagem são rejeitados ou aceites na criação de uma nova imagem. A figura 4.17, é visível que os objetos "incompletos" foram automaticamente descartados na nova imagem.



Figura 4.17: Imagem final do exemplo (a) e (b)

Com este algoritmo também é possível eliminar objetos que possam encontrar-se na imagem, e que não contenham as dimensões padrão anteriormente definidas. Assim, os objetos "muito grandes", "muito pequenos", "muito compridos" ou "muito largos" são eliminados automaticamente.

As imagens da figura 4.17, não cumprem os requisitos mínimos para que seja aplicado o algoritmo *OCR*, como explicado na secção 4.7.

Todavia, um exemplo de leitura bem sucedida, é apresentada na figura 4.18. É visível, que o contador está "quase" a completar a transição de "066 902" para "066 903", mesmo assim não terminou completamente.



Figura 4.18: Contador com transição quase completa

Quando é aplicado o pré-tratamento de imagem que está apresentado na figura 4.19 o ponto (a), é possível verificar que persiste um "pequeno" objeto do algarismo "2", e o objeto "comprido" no canto esquerdo da imagem.



Figura 4.19: (a) Imagem pré-tratada (b) Imagem final adquirida

Todavia, os objetos "pequeno" e "comprido" foram descartados pelo algoritmo, como é visível na figura 4.19 ponto (b). Ambos, não continham as dimensões mínimas aceitáveis e por conseguinte é possível obter uma imagem "limpa" para que seja analisada pelo algoritmo *OCR*.

É visível que o algarismo "3" é aceite na nova imagem, e pode ser analisado pelo algoritmo *OCR*. O que define quais os objetos que são aceites, é controlado por uma variável estabelecida no sistema, como explicado na subsecção 4.6.1.

Neste exemplo utilizou-se o nível de visibilidade a 90%. Após o tratamento de imagem concluído, é aplicado o algoritmo de leitura *OCR* à figura 4.19, ponto (b), o resultado é mostrado na figura 4.20.

```
Leitura->>>>>>: 066903
INSERIR DADOS EM BASE DADOS -> 2
INSERT INTO DADOS (N_LEITURA,DATA,HORA,VALOR) VALUES ('12','May 3 2017','18:39:09','066903')
```

Figura 4.20: Imagem do contador com aplicação de *Threshold*

É visível na figura 4.20, que a leitura foi executada corretamente e ficando automaticamente guardada no sistema do leitor instalado no contador. Todavia, é necessário verificar os valores lidos pelo algoritmo *OCR*.

De forma a aumentar a percentagem de acerto final, aplicou-se à imagem uma operação

morfológica dilatação¹, e três erosões², que pelo ensaio da secção 4.6.1, apresentou um aumento na percentagem de acerto final de 0.2%.

De forma a testar o nível de visibilidade e a influência do tratamento de imagem com erosão e a operação morfológica dilatação, foram executados ensaios com o contador a avançar de forma contínua.

O ensaio teve início com o contador no valor 050 030 e terminou em 050 416, num total de 500 imagens a uma resolução de 1280 * 960 pixels.

As configurações iniciais são apresentadas na tabela 4.4, foram utilizadas em ambos os ensaios.

Tabela 4.4: Configuração inicial aplicada pelo utilizador no ensaio 1

Largura (pixels)	Altura (pixels)	<i>Threshold</i>	<i>Blobscleaner</i>
920	150	automático	857

Como é visível pela tabela anterior, foi utilizada uma *ROI* com 920 * 150 pixels, utilizou-se o *Threshold* adaptativo e foi definido um nível para *Blobscleaner* de 857 pixels.

4.6.1 Nível de visibilidade e acerto final

O ensaio visa determinar que valores para nível de visibilidade permite acertos mais elevados. O gráfico da figura 4.21, apresenta a relação entre a percentagem de acerto e o nível de visibilidade utilizado, num total de 13 ensaios.

O número de ensaios é consideravelmente baixo, porque para cada registo é necessário executar 500 leituras, e verificar a percentagem de acerto final. Este processo, é moroso, e não é executado de forma automática, razão pela qual são apresentadas apenas 13 amostras, mas que representam 6500 leituras efetuadas.

¹Operação morfológica dilatação, onde cada pixel do objeto (valor 1) que esteja ligado a um pixel de fundo (valor 0), coloca esse pixel de fundo a 1

²Operação morfológica erosão, cada pixel do objeto (valor 1) que esteja ligado a um pixel de fundo (valor 0) e coloca esse pixel do objeto a 0

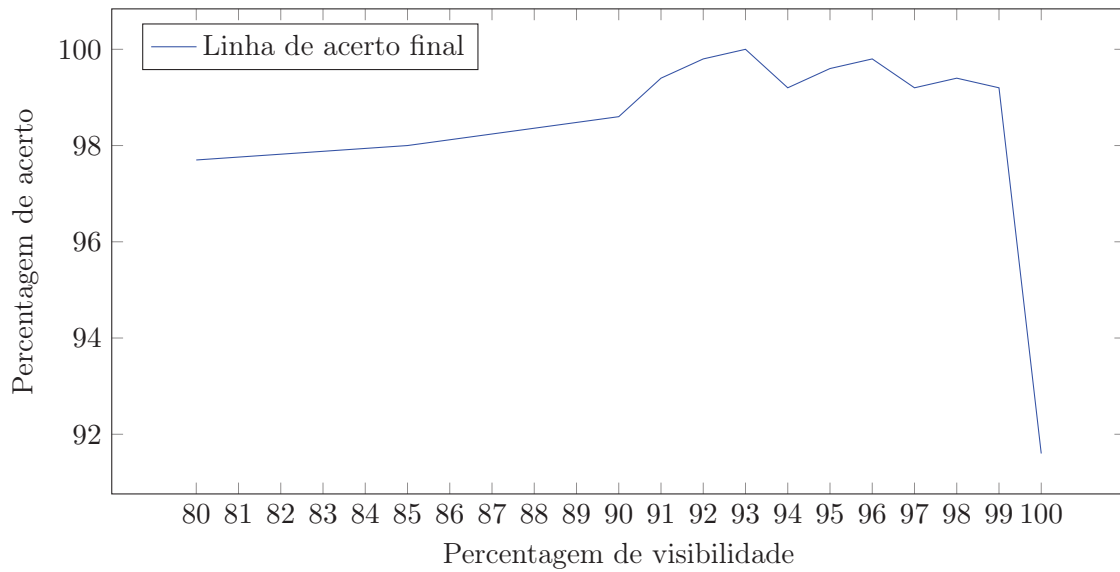


Figura 4.21: Percentagem de acerto e visibilidade

O valor máximo foi adquirido com acerto a 100% na leitura das imagens, e foi atingido com visibilidade a 93%.

4.7 Verificação do valor lido

Uma vez que não é possível garantir que o algoritmo *OCR* obtenha sempre leituras 100% corretas, então desenvolveu-se três processos de pós-processamento que permitem validar a leitura final. Na figura 4.22, está representado o fluxograma do sistema implementado.

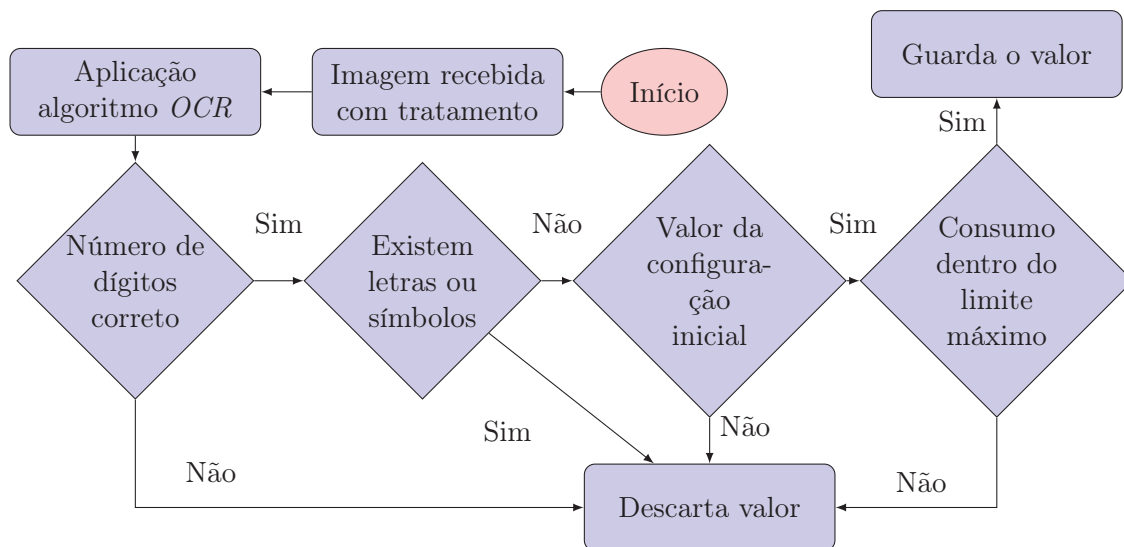


Figura 4.22: Fluxograma de verificação do valor lido pelo algoritmo

O processo inicial consiste na validação do número de dígitos de duas formas distintas.

A primeira compara o número total de dígitos inserido pelo utilizador na configuração inicial, com o valor total lido pelo sistema. Por exemplo, se no contador estiver o valor

003 450, isso significa que tem no total 6 dígitos, que são validados pelo número inserido inicialmente pelo utilizador na configuração inicial.

A segunda forma, consiste na eliminação dos zeros à esquerda do contador. No mesmo exemplo, é tido em consideração o valor 3450, que contém 4 dígitos. De forma a validar o valor lido, é necessário compara-lo com número de dígitos. Para tal, o sistema necessita de executar 10 leituras consecutivas ao contador, e se o número de dígitos não for alterado, então é validado esse mesmo número de dígitos. Por exemplo, quando o contador transita de 9999 para 10 000, o sistema descartará as leituras feitas enquanto não executar 10 leituras consecutivas que permitem atualizar o valor de comparação de 4 para 5, e assim validar o valor lido pelo sistema. Durante este processo, todas as leituras são descartadas, até se confirmar a existência de um novo dígito de comparação.

O segundo processo, passa por verificar se existem letras ou símbolos nos resultados lidos, se tal acontecer, então a leitura é descartada.

O algoritmo *OCR Tesseract*, tem diversas configurações que podem ser aplicadas para aumentar o acerto na leitura. Dentro dessas configurações é possível definir uma lista negra (*blacklist*), que é preenchida com caracteres que não se pretende ter como resultado final.

É possível aplicar também uma lista branca, (*whitelist*), que indica quais caracteres que podem ser lidos. Para a leitura nos contadores analógicos, é unicamente necessário preencher a lista com números de 0 a 9.

O terceiro processo, consiste em verificar se o valor lido é inferior ao valor do contador quando este foi configurado pela primeira vez pelo utilizador, se tal acontecer, a leitura é descartada.

Por último, é feita a verificação do valor lido pelo algoritmo *OCR*, e caso esse valor seja excessivamente alto ou baixo comparativamente ao último lido. Para tal é define-se um intervalo configurável pelo utilizador, que pode depender da instalação. Uma vez a validação dos números executada, é possível utilizar as leituras que foram aprovadas para a deteção de fugas no sistema de abastecimento.

4.8 Deteção de fugas

Os alarmes têm o propósito de alertar o utilizador sobre consumos anormais registados nas redes de abastecimento, estes podem ser divididos em dois grupos. O primeiro incide nos consumos em habitações particulares e o segundo sobre os registos nas redes de abastecimento.

A deteção de fugas em habitações, explicadas na secção 2.12, pode ser efetuada através da medição do fluxo de água num determinado período de tempo.

Na proposta desenvolvida, esta é dividida em ciclo curto, diário e semanal. É possível também classificar as fugas em alta ou baixa intensidade e para tal, são utilizadas duas variáveis, as diferenças de "zero" e "máximos" dos consumos.

Portanto, as fugas de baixa intensidade são detetadas com a contagem de vezes que a diferença nos resultados lidos é zero, e para alta, pelos valores que ultrapassam um máximo estabelecido.

Relativamente aos ciclos, o curto é executado de cada vez que é feita uma nova leitura ao contador, e é verificado a existência de fuga de alta intensidade, através da diferença do valor anterior com o atual, se esta ultrapassar um determinado limite definido pelo utilizador, então é emitido um alarme por "máximo".

O ciclo diário, é executado no fim de um período de 24 horas, e é somado o número vezes que a diferença nos valores deu zero. Caso este somatório seja inferior a um determinado limite,

igualmente definido pelo utilizador, então é ativado o alarme para fuga de baixa intensidade.

Por último, o ciclo semanal, verifica ambos os acontecimentos, ou seja, fugas de alta e baixa intensidade, num período de sete dias.

O fluxograma 4.23, apresenta o sistema desenvolvido para deteção de fugas na habitação particular.

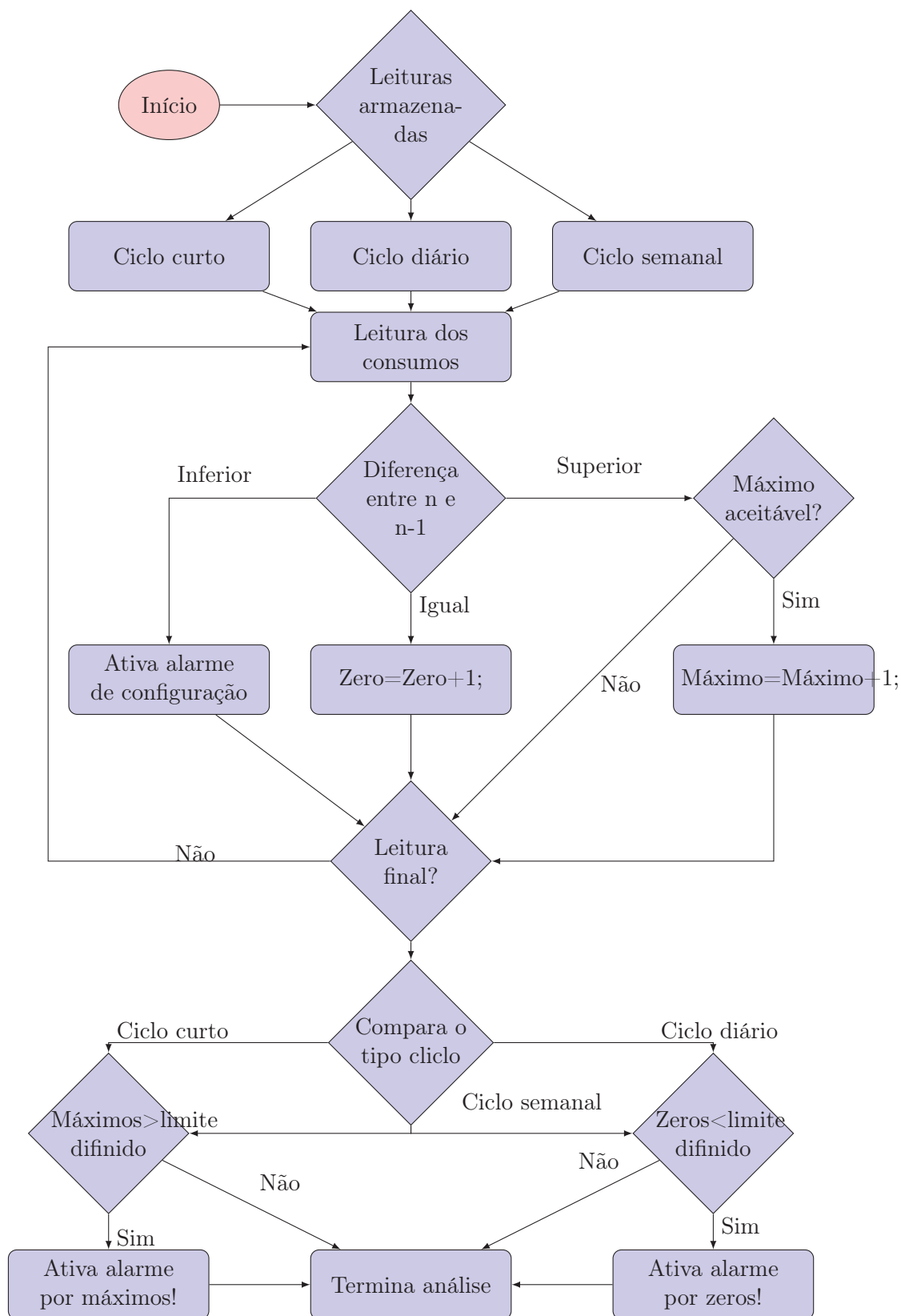


Figura 4.23: Fluxograma geral para a deteção de fugas no leitor

No fluxograma apresentado, caso a diferença seja inferior poderá significar que o contador

esteja a fazer leituras erradas e, portanto, ativa um alarme que permite dar alerta ao utilizador.

Estas podem ser detetadas pela diferença dos consumos efetuados pelos contadores. A figura 4.24, apresenta o fluxograma proposto para deteção de fugas nas redes de abastecimento.

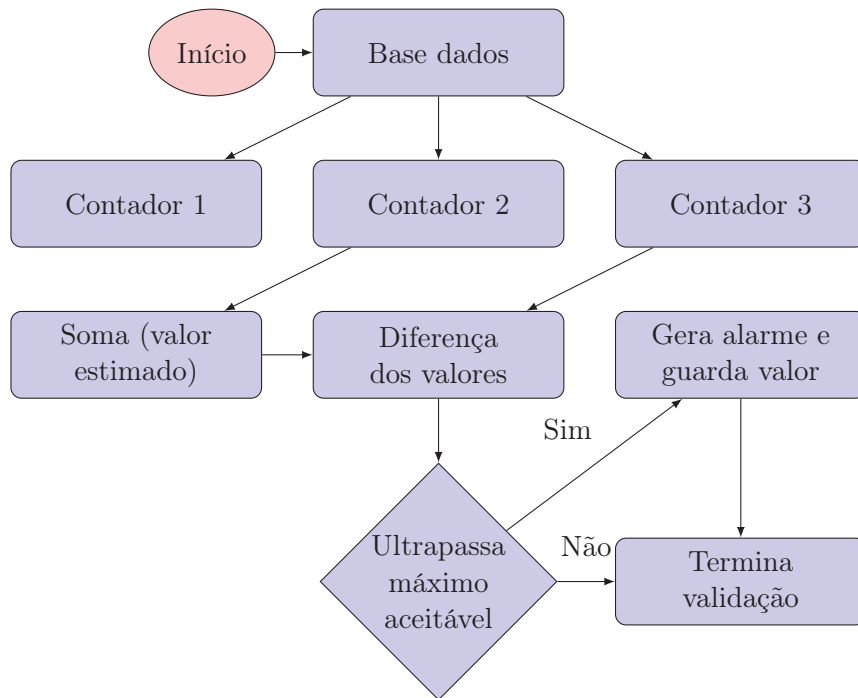


Figura 4.24: Fluxograma geral para deteção de fugas no servidor

Onde o valor Contador 3 é a soma do Contador 1 com o Contador 2, como representado na figura 4.25. Neste exemplo, o valor da soma do Contador 1 com o Contador 2, representa o valor estimado, que se espera obter na leitura do Contador 3 (valor real).

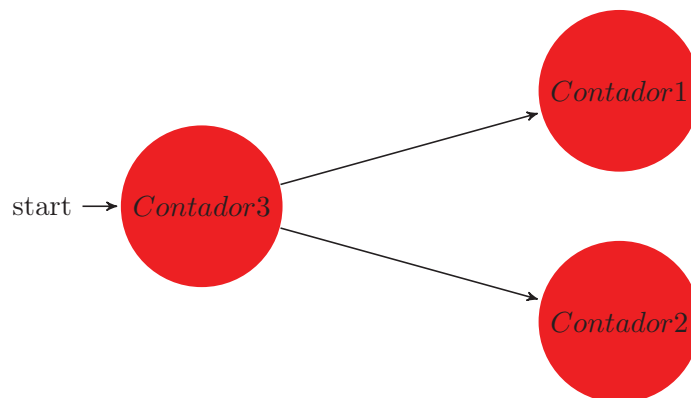


Figura 4.25: Relação entre os contadores instalados na rede de abastecimento

Se a diferença entre o valor real (leitura do Contador 3) e o estimado (soma do Contador 1 com Contador 2), ultrapassar um determinado valor máximo, então é emitido um alarme que alerta o utilizador.

4.9 Fluxograma do sistema em modo autónomo

O leitor instalado no contador analógico, entra em modo autónomo quando a configuração inicial estiver devidamente finalizada. Neste modo, o leitor executa leituras ao contador analógico autonomamente. O número de leituras é determinado pelo utilizador e assim que o valor é atingido, envia os dados para o *Back-end*.

Neste modo de leitura, se o contador analógico que se pretende ler, estiver num processo de transição. Então o leitor irá executar um número de tentativas pré-estabelecidas e desliga-se agendando uma nova leitura. No fluxograma 4.26, é apresentado o fluxo de dados em modo autónomo do sistema.

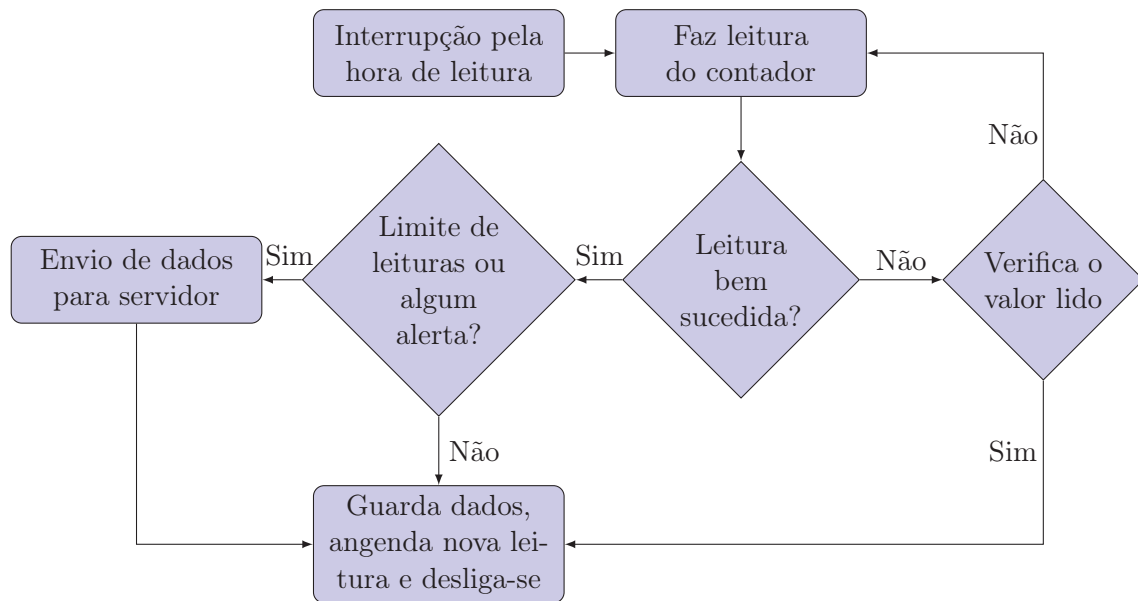


Figura 4.26: Fluxograma do algoritmo em modo autónomo

Tal como explicado na secção 4.4, na verificação do valor lido, se houver por exemplo dez tentativas de leitura falhadas, então é agendada uma nova leitura sem que o algoritmo tenha obtido um valor do contador válido.

4.9.1 Gestão de dados em modo autónomo

No sistema desenvolvido aplicado ao leitor encontram-se dois modos de armazenamento de dados. O primeiro refere-se aos dados de pré-configuração, que são primordiais, pois, têm de ser consultados invariavelmente quando o sistema inicia uma nova leitura em modo autónomo. Na tabela 4.5, estão apresentados cada um desses dados.

Tabela 4.5: Dados de pré-configuração em modo autônomo

Ordem	Função
1	Coordenada X da <i>ROI</i> na imagem
2	Coordenada Y da <i>ROI</i>
3	Comprimento da <i>ROI</i>
4	Largura da <i>ROI</i>
5	Valor do <i>Threshold</i> (só em modo manual)
6	Valor da <i>BlobsCleaner</i> selecionado
7	Número lido inicialmente no contador no momento da configuração
8	Número de dígitos do contador analógico
9	Número de dígitos da última leitura <i>OCR</i>
10	Número total de leituras efetuadas

O segundo modo refere-se aos dados das leituras efetuadas ao contador analógico, que são armazenadas até serem enviadas para o servidor. Para proceder ao envio dos dados definiu-se uma trama, apresentada na tabela 4.6.

Tabela 4.6: Dados enviados do leitor para o *Back-end*

Ordem	Identificador e função
Dado 1	A seguido número da leitura
Dado 2	B seguido valor do consumo do contador
Dado 3	C seguido ano
Dado 4	D seguido mês
Dado 5	E seguido dia
Dado 6	F seguido hora
Dado 7	G seguido minuto
Dado 8	H seguido segundo
Dado 9	I seguido alarme

Quando o leitor instalado no contador executa uma leitura com êxito, é criado (caso não exista no sistema) um ficheiro onde são guardadas as leituras. Por exemplo, supondo que uma leitura com a seguinte trama: A4,B4563,C2017,D08,E20,F18,G30,H50,I1. Neste exemplo a leitura foi enumerada com o número 4, tem valor de 456,3 m^3 , foi executada na data de, 2017-08-20 às 18:30:50 horas e como o "I" é seguido de um "1", então indica uma possível fuga no sistema. O ficheiro vai guardar as diversas leituras, até que seja necessário enviá-lo para o servidor.

Capítulo 5

Testes e validações

Este capítulo tem a finalidade apresentar e analisar os resultados dos testes efetuados ao *software* desenvolvido, através de ensaios práticos. Estes consistem em testar o sistema de visão, detecção de fugas em habitações e redes de abastecimento, obtenção de máximos, mínimos, totais dos consumos efetuados e medição da velocidade na execução do código.

5.1 Teste ao sistema de leitura

Dos ensaios ao sistema de visão, só os de superior relevância são apresentados nesta secção. Estes podem ser executados de duas formas distintas. A primeira consiste no movimento contínuo (velocidade fixa) e uniforme do contador analógico.

O segundo consiste em colocar o contador analógico em movimento com velocidade variável, com o propósito de simular um consumo em um ambiente real.

Para cada ensaio efetuado, são apresentadas as configurações iniciais. Estas apresentam a largura e altura em pixels da *ROI*, *Threshold* e *Blobscleaner* aplicado.

São igualmente mostrados os dados sobre o contador analógico, relativamente à posição inicial e final do ensaio e a número de imagens capturadas no processo.

As imagens capturadas são divididas em imagens bem sucedidas e descartadas. Através destas, é possível quantificar a percentagem de acerto final.

Para compreensão dos ensaios executados, são apresentados dois gráficos distintos. O primeiro consiste em mostrar o resultado da leitura e o segundo em apresentar as diferenças dos valores do primeiro gráfico. Neste último é possível analisar com superior detalhe os erros de leitura no sistema de visão. Pois, um valor negativo no gráfico, provém da diferença das leituras seguinte com a anterior. Logo é possível concluir que existiu um erro na leitura aplicada ao contador, pois, o valor apresentado pelo contador analógico é exclusivamente crescente. Por último, são apresentadas as alterações aplicadas ao programa, de forma a aumentar a percentagem de acerto final, resumidas na tabela 5.1.

Tabela 5.1: Tipo de alterações executadas no código ao longo dos ensaios

Definição	Simplificação	Tipo de alteração
Tipo de <i>Threshold</i>	(TT)	Manual ou automática
Configuração do <i>Tesseract</i>	(CT)	Aplicado ou não
Porcentagem mínima para a <i>BondingBox</i>	(PMBB)	90% ou 93%
Verificar o número de dígitos no contador	(VNDC)	Aplicado ou não
Verificação de letras ou símbolos	(VLS)	Aplicado ou não
Verifica o valor mínimo do contador	(VMC)	Aplicado ou não
Verifica dentro do limite definido	(VLD)	Aplicado ou não
Resolução de imagem da câmara	(RIC)	800*600 ou 1290*960 pixels
Aplicação da dilatação e erosão na imagem	(DEI)	Aplicado ou não

As alterações ao programa são apresentadas na subsecção 4.5.2 e secções 4.6 e 4.7. Destas apenas a alteração da resolução não foi abordada, isto acontece porque só na fase final dos ensaios houve alteração da resolução de 800 * 600 pixels para 1290 * 960 pixels. Esta alteração melhorou significativamente a percentagem de acerto final, como demonstrado no ensaio da subsecção 5.1.4.

5.1.1 Ensaio ao sistema de visão 1

O primeiro e segundo ensaios foram executados com o contador a avançar de forma contínua. O contador analógico iniciou o valor em 0 e terminou em 3000, num total de 1276 imagens com resolução de 800 * 600 pixels. As configurações iniciais apresentadas na tabela 5.2, foram as mesmas para ambos os ensaios.

Tabela 5.2: Configuração inicial aplicada pelo utilizador no ensaio 1

Largura (pixels)	Altura (pixels)	Threshold	Blobscleaner
264	63	40	65

Nesta primeira leitura, apenas foi definido o nível mínimo de visibilidade a 90%, como apresentado na tabela 5.3.

Tabela 5.3: Alterações feitas ao sistema no ensaio 1

TT	CT	PMBB	VNDC	VLS	VMC	VLD	RIC (pixels)	DEI
Manual	Não	90%	Não	Não	Não	Não	800*600	Não

O gráfico apresentado na figura 5.1, apresenta a evolução das leituras do contador analógico. Verifica-se a existência de diversos erros na leitura aplicada ao contador. O ensaio foi executado em modo de velocidade fixa, logo a linha apresentada deveria ser reta contínua e não apresentar variações de "picos" e "vales". Estes valores traduzem-se em erros de leitura, que influenciam a percentagem de acerto final.

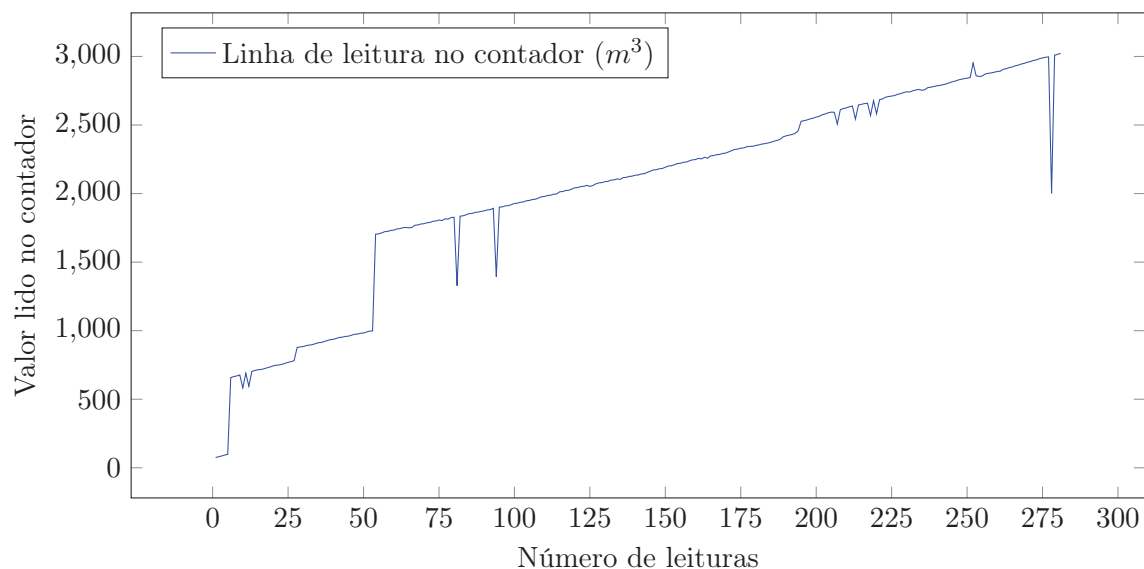


Figura 5.1: Leituras desde 0 até 3000 no contador em 1276 imagens

Pelo gráfico das diferenças apresentadas na figura 5.2, é possível verificar que em diversos pontos os valores foram negativos e que se traduziram em erros de leitura do sistema. A seguir aos "picos" negativos, existem os "picos" de acerto do valor lido.

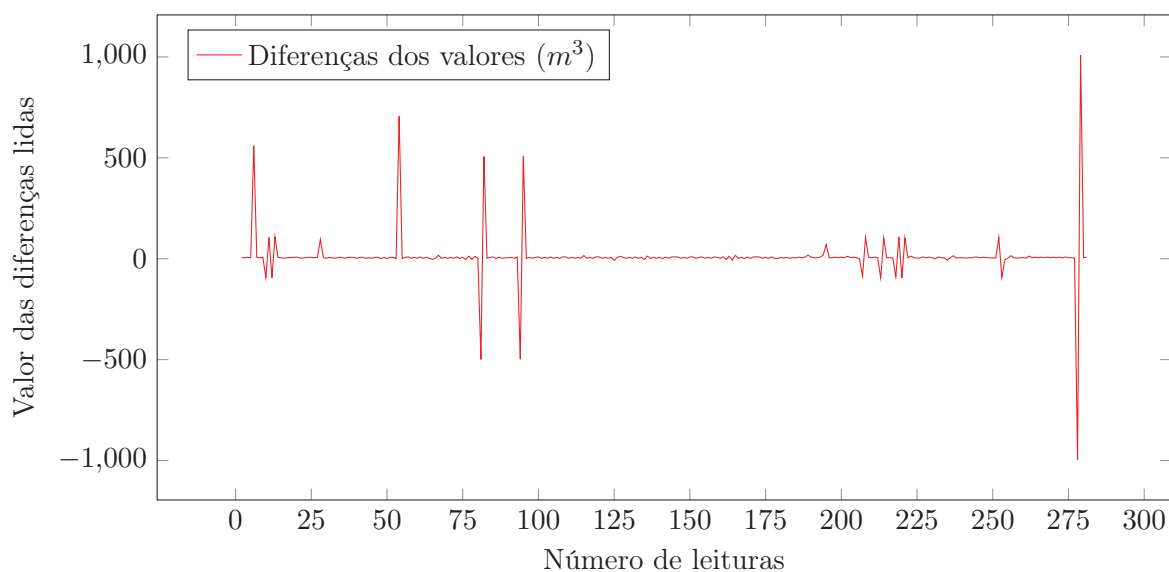


Figura 5.2: Diferenças entre os números lidos do ensaio 1

A tabela 5.4, apresenta resultado do ensaio prático. É possível verificar que, foram executadas 260 leituras bem sucedidas e 995 foram descartadas pelo sistema, ou seja, das 1276 imagens, 77,97% correspondem a imagens descartadas pelo sistema.

Tabela 5.4: Resultados do ensaio 1

Modo	Início - fim	Total	Bem sucedidas	Descartadas	Erradas	Acerto (%)
Fixo	0-3000	1276	260	995	21	91,92

É necessário ter em consideração que 77,9% do total das imagens foram descartadas. Este valor depende de inúmeras variáveis, como a velocidade a que o contador analógico apresenta os dados, o número de fotografias tiradas pela câmara no ensaio, e da validação de dados, como explicado na secção 4.7. Deste modo, das 260 bem sucedidas 21 foram leituras erradas, levando o acerto final de 91,92%.

5.1.2 Ensaio ao sistema de visão 2

No segundo ensaio, comparativamente ao primeiro, acrescentou-se a verificação de número de dígitos no contador e letras ou símbolos nos valores lidos. A tabela 5.5, apresenta o resumo das alterações feitas ao programa.

Tabela 5.5: Alterações feitas ao sistema do ensaio 2

TT	CT	PMBB	VNDC	VLS	VMC	VLD	RIC (pixeis)	DEI
Manual	Não	90%	Sim	Sim	Não	Não	800*600	Não

Pelo gráfico apresentado na figura 5.3, é possível verificar comparativamente ao ensaio anterior uma melhoria no resultado final.

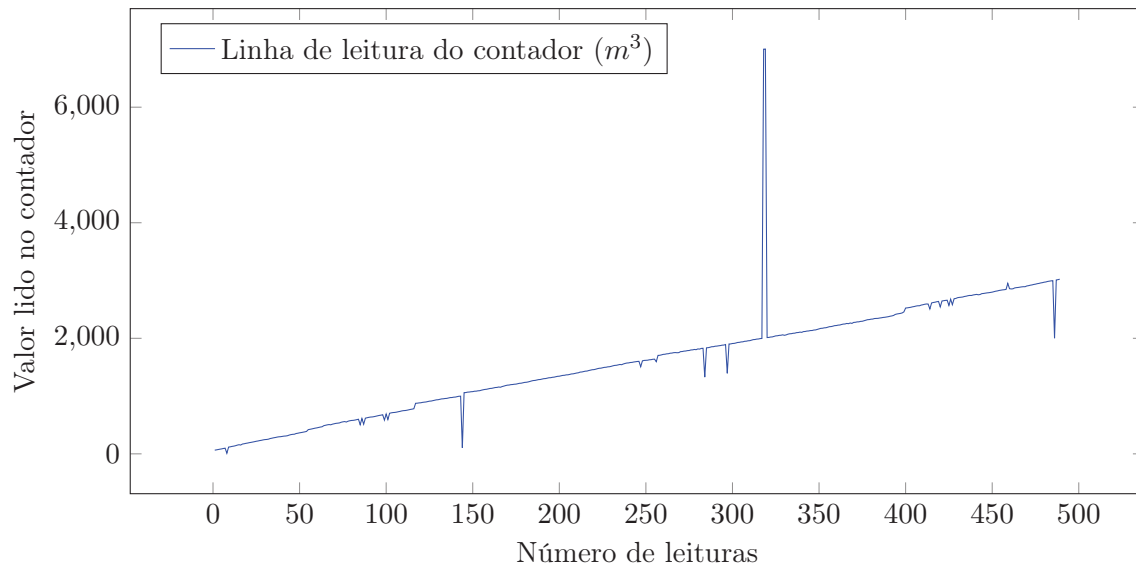


Figura 5.3: Leituras desde 0 até 3000 no contador em 1276 imagens

Pelo gráfico apresentado na figura 5.4, é possível verificar a evolução, contudo permanecem erros relevantes na leitura.

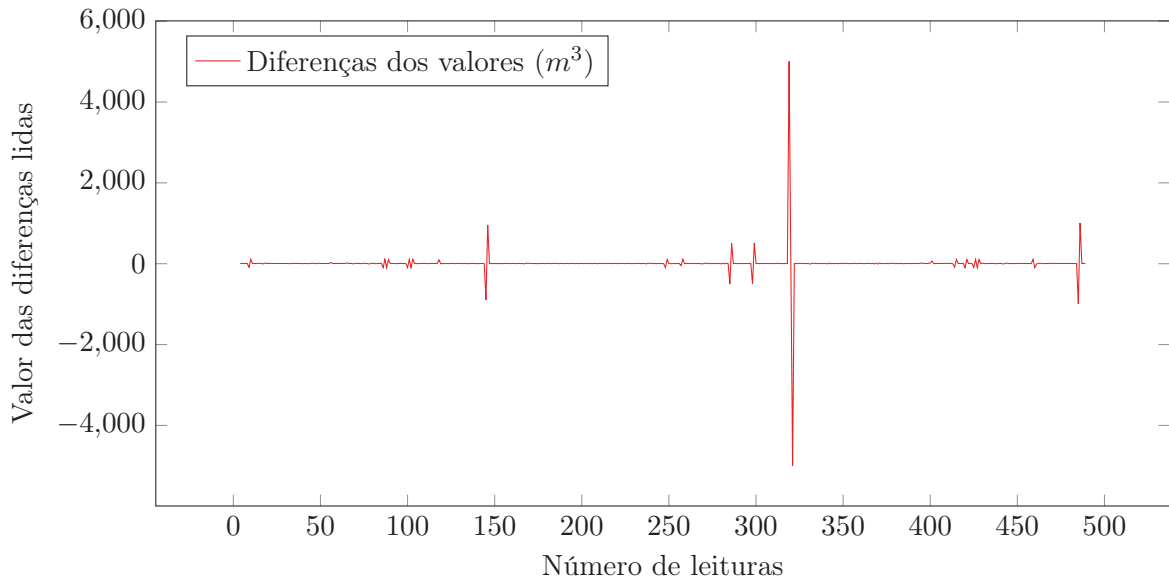


Figura 5.4: Diferenças entre os números lidos do ensaio 2

A tabela 5.6 apresenta os dados resumidos neste ensaio. Das 1276 imagens analisadas, 456 foram leituras bem sucedidas e 787 descartadas, ou seja 61.67% das imagens.

Tabela 5.6: Resultados do ensaio 2

Modo	Início - fim	Total imagens	Bem sucedidas	Descartadas	Erradas	Acerto (%)
Fixo	0-3000	1276	456	787	33	92,76

Com as alterações feitas ao programa, verificou-se um aperfeiçoamento na percentagem de acerto de 91,92% para 92,76%. Independentemente de ser um aperfeiçoamento pouco relevante, é necessário ter em consideração que o número de imagens aumentou em 16,3%. O que permite concluir que o sistema ficou mais estável nas leituras executadas.

5.1.3 Ensaio ao sistema de visão 3

O terceiro ensaio, executou-se com o contador a avançar com velocidade não uniforme (descontínua), onde se variou a velocidade de contagem de modo aleatório. Assim o contador avançou da posição inicial de valor 57 452 até 57 709 num total de 920 imagens com resolução de 800 * 600 pixels.

Neste ensaio, apresentado na tabela 5.7, alterou-se o tipo *Threshold* de manual para automático.

Tabela 5.7: Configuração inicial aplicada pelo utilizador no ensaio 3

Largura (pixels)	Altura (pixels)	Blobscleaner
264	63	65

As alterações apresentadas na tabela 5.8, resumem-se à configuração do algoritmo *OCR*

Tesseract, na verificação do valor mínimo e na confirmação do valor dentro de um limite pré-determinado.

Tabela 5.8: Alterações feitas ao sistema no ensaio 3

TT	CT	PMBB	VNDC	VLS	VMC	VLD	RIC (pixels)	DEI
Automático	Sim	93%	Sim	Sim	Sim	Sim	800*600	Não

É visível no gráfico da figura 5.5, a linha apresentada tornou-se mais estável. É necessário ter em conta que este ensaio é executado com o contador em velocidade variável.

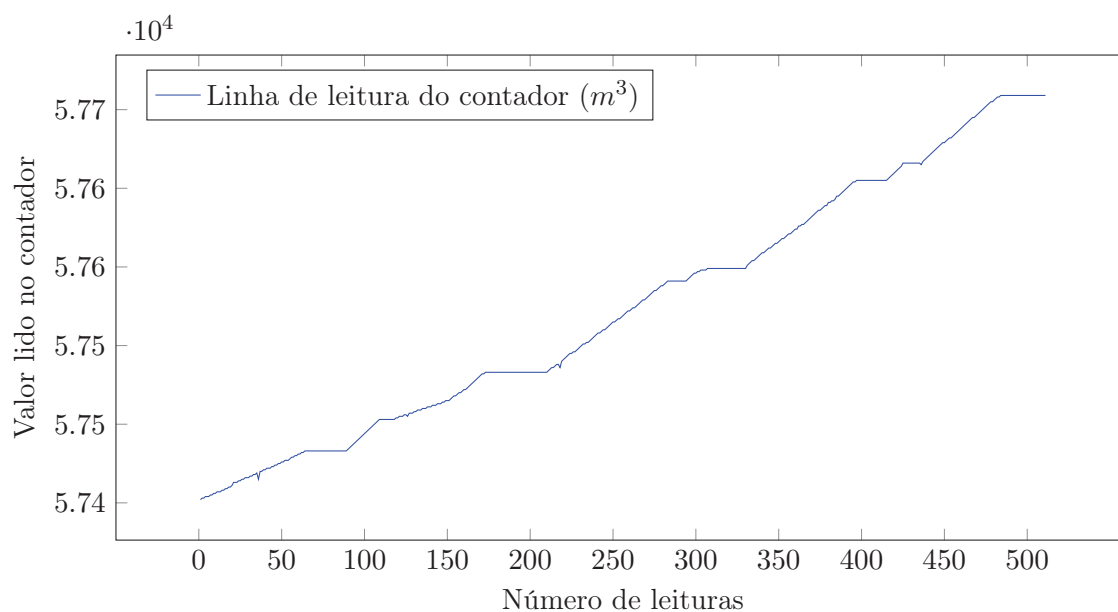


Figura 5.5: Leituras desde 57452 até 57709 em 920 imagens

No gráfico da figura 5.6, é possível verificar que existiu aumento de estabilidade nos valores lidos, porém ainda com alguns erros de leitura.

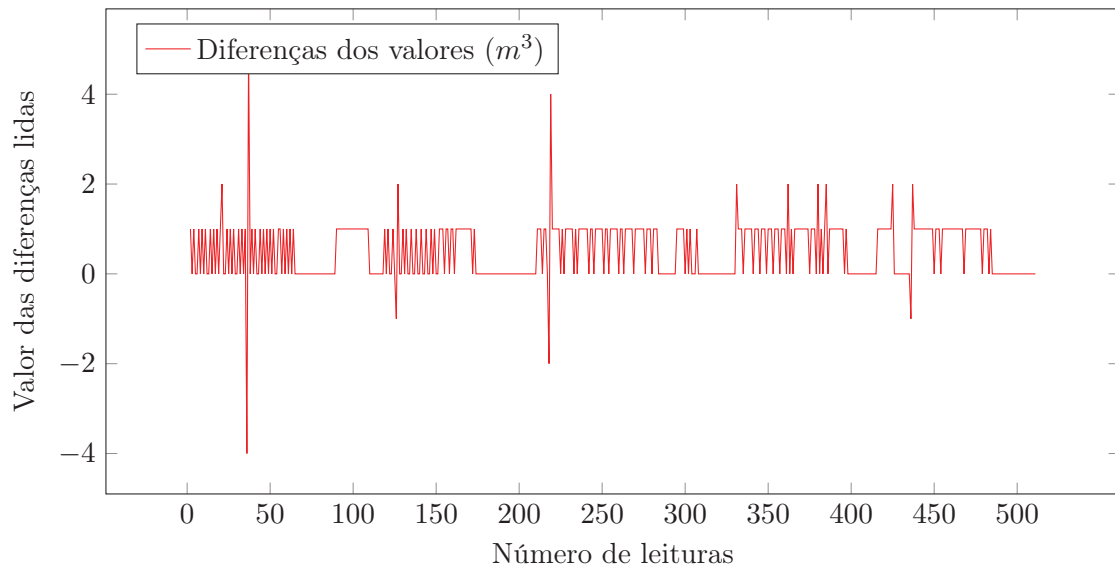


Figura 5.6: Diferenças entre os números lidos no ensaio 3

Neste ensaio, as alterações ao sistema permitiram subir a percentagem de acerto para 99,57%, no modo com velocidade variável. Onde a percentagem de imagens descartadas foi de 44,02%. Todavia, este valor não pode ser comparado com os anteriores, pois, varia conforme as imagens e com a velocidade de contagem do próprio contador analógico. A tabela 5.9, apresenta o resumo dos resultados neste ensaio.

Tabela 5.9: Resultados do ensaio 3

Modo	Início - fim	Total	Bem sucedidas	Descartadas	Erradas	Acerto (%)
Variável	57452-57709	920	511	405	4	99,22

Estas alterações permitiram subir comparativamente ao último ensaio a percentagem de acerto em 6,46%, para 99,22%. Esta subida na percentagem de acerto deve-se sobretudo à configuração aplicada ao algoritmo *OCR Tesseract*, explicada na secção 4.5.2. Outro fator relevante deve-se ao aumento do nível de visibilidade de 90% para 93%, explicada na secção 4.6.

5.1.4 Ensaio ao sistema de visão 4

Este ensaio foi executado em modo de velocidade fixa, onde o contador variou de 50 031 até 56 113 num total de 8000 imagens. A resolução da imagem, foi aumentada de 800 * 600 para 1280 * 960 pixels.

Consequentemente as dimensões da *ROI*, foram definidas em 920 pixels de largura e 180 pixels de altura. O valor do *Blobscleaner* foi igualmente configurado em 845. A tabela 5.10 apresenta o resumo dos dados configurados inicialmente pelo utilizador.

Tabela 5.10: Configuração inicial aplicada pelo utilizador no ensaio 4

Largura (pixeis)	Altura (pixeis)	Blobscleaner
920	180	845

As alterações ao programa, apresentadas na tabela 5.11, consistiram no aumento da resolução e na aplicação das operações morfológicas de dilatação e erosão à imagem final.

Tabela 5.11: Alterações feitas ao sistema no ensaio 4

TT	CT	PMBB	VNDC	VLS	VMC	VLD	RIC (pixeis)	DEI
Automático	Sim	93%	Sim	Sim	Sim	Sim	1280*960	Sim

Pelo gráfico da figura 5.7, é possível verificar uma linha contínua nas 8000 imagens.

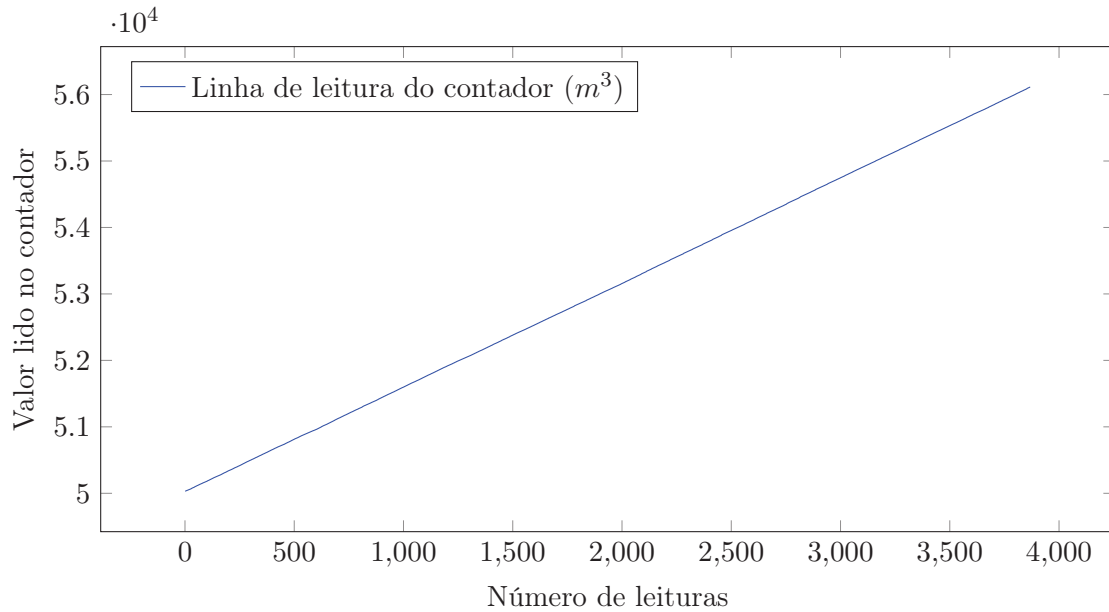


Figura 5.7: Leituras desde 50031 até 56113 em 8000 imagens

Pelo gráfico da figura 5.8, é possível verificar que não houve números negativos. Portanto, neste ensaio não foram detetados erros de leitura durante o processo.

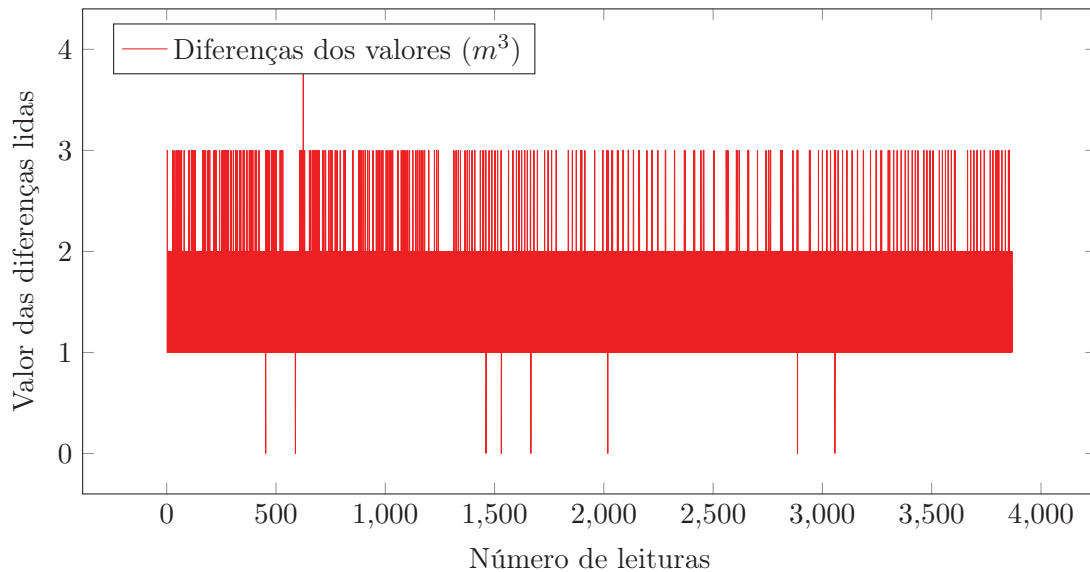


Figura 5.8: Diferenças entre os números lidos no ensaio 4

Pela tabela 5.12, é possível ver que das 8000 imagens, 3868 foram leituras bem sucedidas e 4132 descartadas ou seja, 51,65% das imagens. É igualmente possível verificar que não existiram imagens erradas, e logo 100% de acerto.

Tabela 5.12: Resultados do ensaio 4

Modo	Início - fim	Total	Bem sucedidas	Descartadas	Erradas	Acerto (%)
Fixo	50031-56113	8000	3868	4132	0	100

A percentagem de acerto, deve-se principalmente ao aumento da resolução de imagem de 800 * 600 pixels para 1280 * 960 pixels e à aplicação das operações morfológicas.

5.2 Simulação de fuga numa habitação

Para além dos ensaios ao sistema de visão, é necessário fazer testes ao sistema para deteção de fugas na habitação, que é explicado na secção 2.12. Para tal, foi executado um ensaio em modo com velocidade variável, cujo contador iniciou no valor 57 109 até 57 651 em 906 imagens por um período de 10 horas. O ensaio prolongou-se durante este tempo, para que fosse mais próximo de uma aplicação real.

Neste ensaio, são apresentados os valores de consumo total, máximo, mínimo e a possibilidade de fuga no sistema. Para tal, é estabelecido um limite máximo de consumo num determinado período de tempo, neste ensaio as imagens foram capturadas em intervalos de aproximadamente de 2 minutos. Estabeleceu-se um consumo máximo de sensivelmente 0,1 m^3/min .

As condições do ensaio aplicadas, foram as mesmas que do ensaio 4, o gráfico apresentado na figura 5.9 mostra a evolução do consumo efetuado.

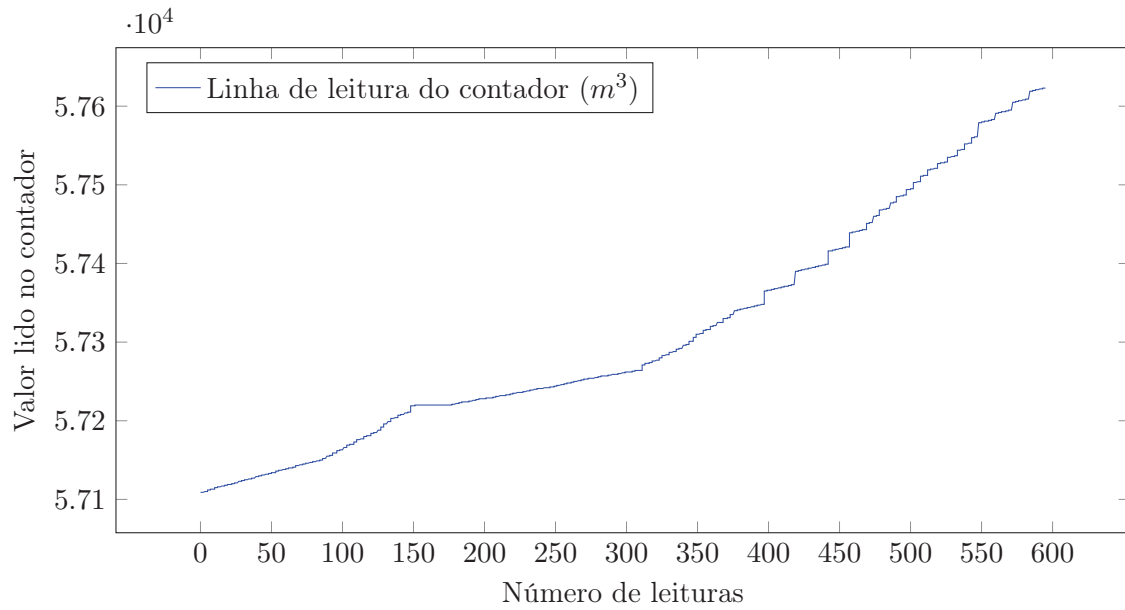


Figura 5.9: Leituras desde 57109 até 57651 em 906 imagens

Pelo gráfico das diferenças da figura 5.10, é possível verificar que não foram detetados erros no ensaio. É igualmente possível verificar que em diversas vezes que o contador não esteve em movimento, onde a diferença do valor atual com o anterior é zero.

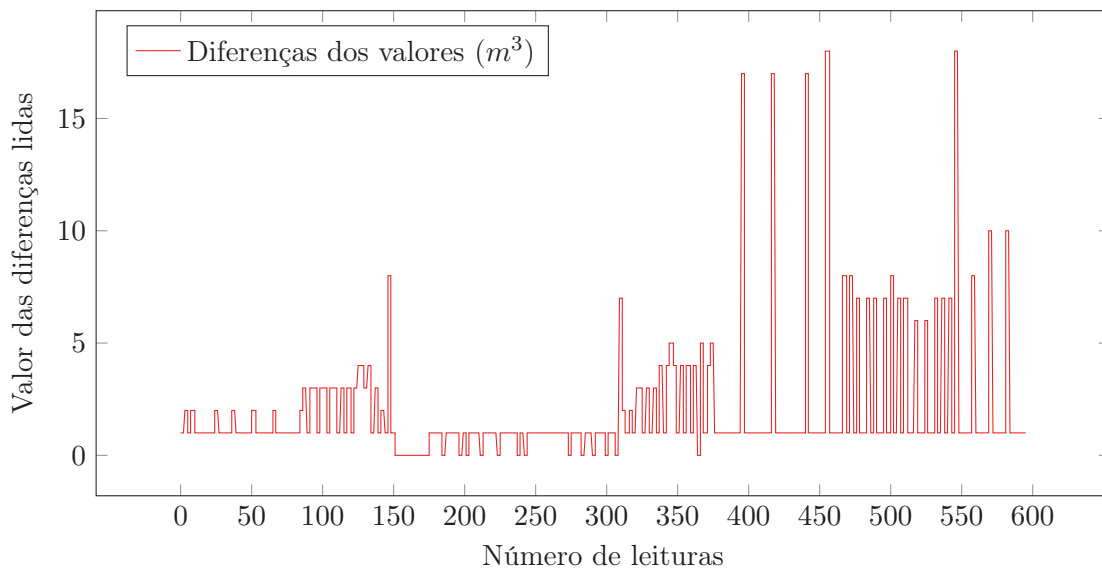


Figura 5.10: Diferenças entre os números lidos da simulação

Pela tabela 5.13, é possível verificar que das 906 imagens analisadas, 804 foram bem sucedidas e 102 descartadas. Das imagens analisadas, não foram registados erros de leitura o que consistiu num acerto de 100%.

Tabela 5.13: Resultados da simulação

Modo	Início - fim	Total	Bem sucedidas	Descartadas	Erradas	Acerto (%)
Variável	57109-57651	906	804	102	0	100

Na tabela 5.14, é igualmente possível verificar o consumo total, máximo e mínimo do ensaio.

Tabela 5.14: Dados sobre os consumos no ensaio

Modo	Consumo total (m^3)	Mínimo (m^3)	máximo (m^3)
Fixo	51,4	0	18

O gráfico apresentado na figura 5.11, apresenta os consumos que foram superiores a $0,1m^3/min$, ou seja, que apresentam um consumo acima de um determinado valor estabelecido. Neste gráfico se o valor ultrapassa os $0,1m^3/min$, é gerado um alarme.

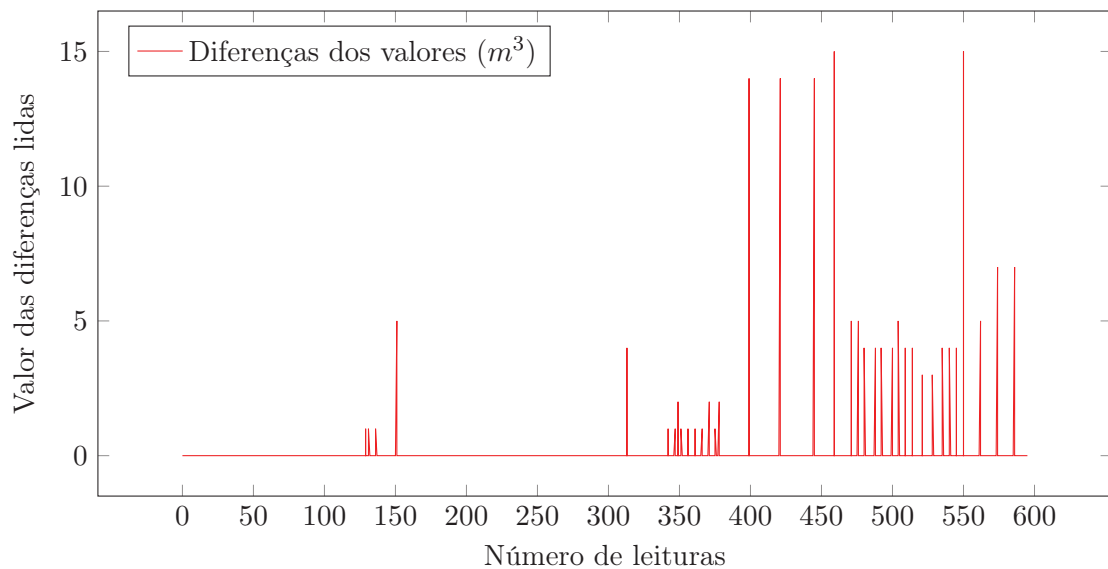


Figura 5.11: Consumos registados acima do limite definido

Pela tabela 5.15, é possível concluir que 37 passagens ocorreram acima do limite de $0,1m^3/min$ num total de $17,3 m^3$ consumidos, acima do valor limiar.

Tabela 5.15: Resultados da análise do teste do sistema

Modo	Limite (m^3/min)	Passagens do limite	Total acima do limite (m^3)
Deteção de fugas	0,1	37	17,3

5.3 Simulação de fuga em rede de abastecimento

Dado que não foi possível construir três protótipos da unidade de leitura, optou-se por simular consumos e analisar os resultados.

Supondo uma rede de abastecimento constituída por 3 contadores, que como referido na secção 4.8, a soma do contador 1 com o contador 2, representa o valor estimado e o valor do contador 3 apresenta o valor real medido.

De forma a simular uma fuga na rede de abastecimento, os dados do contador 1 foram alterados, como apresentado no gráfico 5.12.

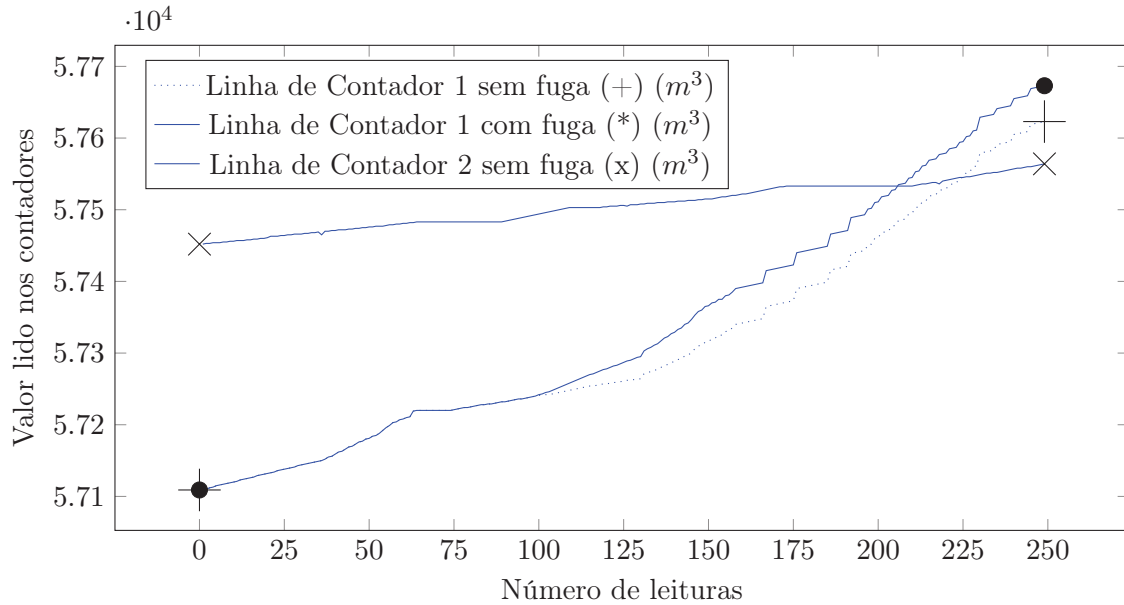


Figura 5.12: Simulação de fuga na rede de abastecimento

Neste exemplo, os dados do contador 2 não foram alterados, assim, a soma do contador 1 com o contador 2, será diferente do valor lido no contador 3. Desta forma, são visíveis três linhas de consumos, a linha do contador 1 com e sem fuga, e a linha do contador 2 sem fuga.

Assim, é possível comparar os valores estimados com os medidos no contador 3. A figura 5.13, apresenta o valor estimado e medido, como é visível existe uma diferença nos valores medidos.

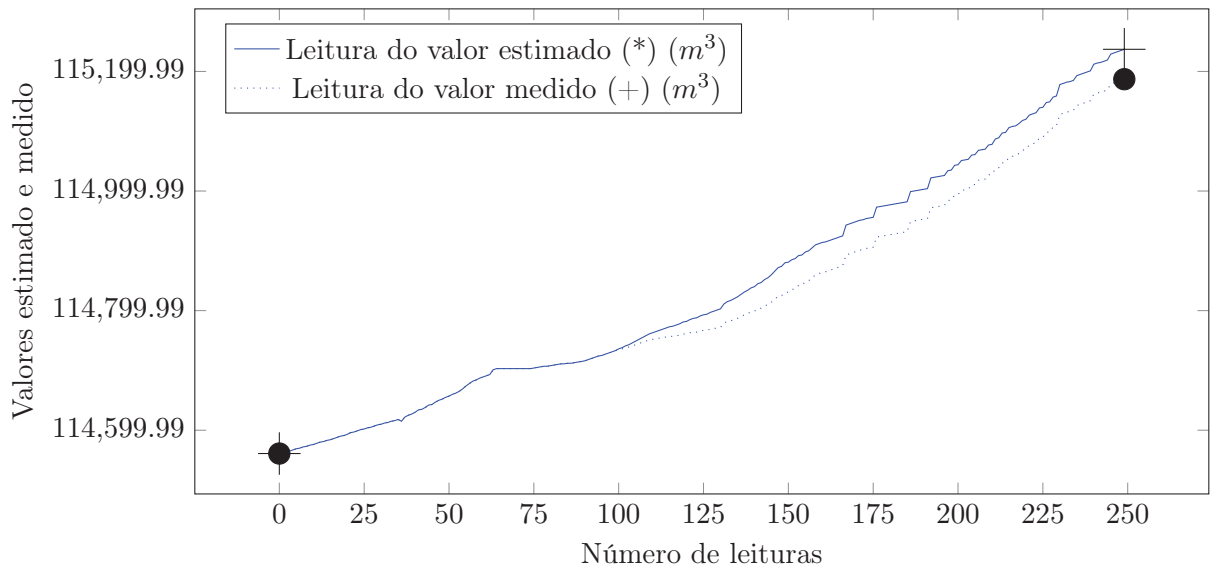


Figura 5.13: Leitura estimada e medida

A figura 5.14, mostra a diferença entre o consumo medido e estimado. É visível, a partir da leitura número 100, uma subida nos valores lidos. Esta alteração, é indicadora de alguma fuga na instalação e pode ser utilizada para gerar um alarme caso ultrapasse um determinado valor pré-estabelecido.

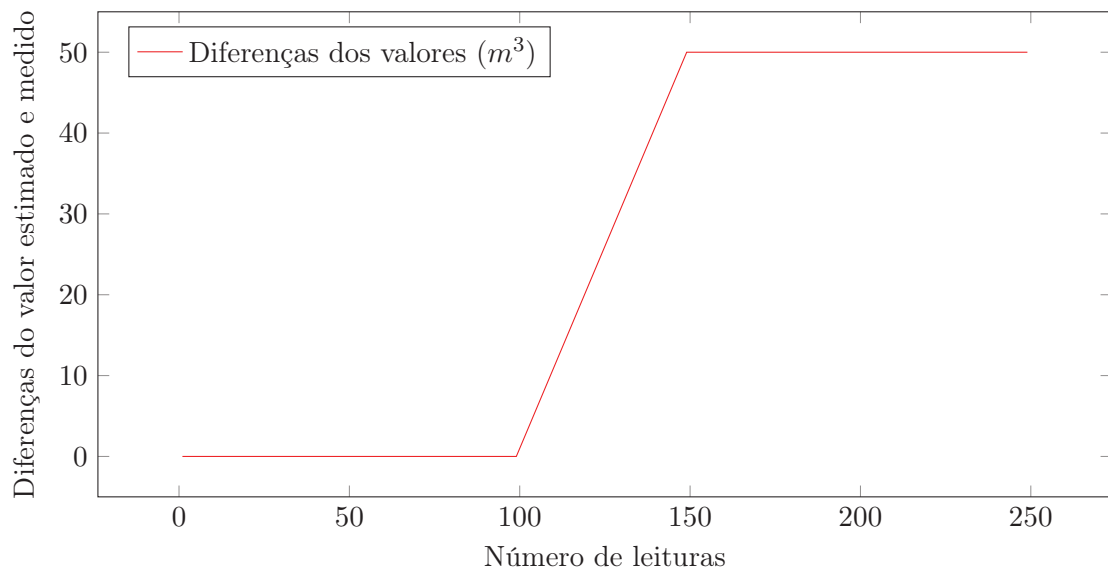


Figura 5.14: Diferenças entre os valores estimado e medido

Neste ensaio, foram adicionados m^3 cúbicos de água aos valores medidos pelo leitor 1. Neste exemplo a fuga teve início na leitura número 100, e foi incrementado até à contagem 150, onde posteriormente permaneceu constante até ao final do ensaio. A tabela 5.16, apresenta a perda total desde a leitura número 100.

Tabela 5.16: Resultados da análise do teste do sistema

Início	Fim	Perda total (m^3)
100	150	1127,5

O valor da perda total é obtido pela diferença dos valores consumos e valores estimados.

5.4 Tempo de ciclo do programa

Esta secção tem finalidade testar o código relativamente à velocidade de processamento. O propósito destes testes é medir o tempo de processamento de cada leitura, e compara-lo com os tempos se fosse aplicado à placa de desenvolvimento *Raspberry Pi 3*. É possível igualmente determinar quantas leituras é possível executar num determinado período de tempo.

Este ensaio divide-se em duas partes, a primeira consiste no teste do código sem que a aquisição seja feita pela câmara e o segundo em testar o código com a aquisição de imagem executada pela câmara. Esta distinção é feita, porque o tempo de aquisição de imagem varia de câmara para câmara.

Neste ensaio foi utilizado um computador com as características apresentadas na seguinte tabela 5.17.

Tabela 5.17: Dados de teste do sistema

Marca	Processador	Memória RAM
HP	i5-6200 @2,3GHz*4	6 GB

5.4.1 Tempo de execução de código

Este ensaio consiste na execução do código sem o tempo de captura de imagem por parte da câmara. Para isso adquiriram-se previamente imagens que foram guardadas no disco do computador para depois serem acedidas pelo programa de leitura.

No ensaio foram seleccionadas aleatoriamente cinco imagens, e para cada imagem executou-se o código dez vezes. Na tabela 5.18, estão os dados registados.

Tabela 5.18: Ensaio ao teste de velocidade feito a cinco imagens diferentes

Nº	img1 (s)	img2 (s)	img3 (s)	img4 (s)	img5 (s)
1	0,255374	0,255371	0,256486	0,259875	0,261021
2	0,256819	0,259729	0,256986	0,261704	0,257422
3	0,257002	0,258430	0,259878	0,262529	0,258120
4	0,256077	0,259647	0,258223	0,259348	0,260275
5	0,255022	0,259778	0,256008	0,260254	0,256436
6	0,261431	0,258581	0,256945	0,260430	0,260352
7	0,256998	0,260479	0,257846	0,260801	0,257646
8	0,256023	0,259661	0,255704	0,260567	0,259190
9	0,256761	0,256799	0,257984	0,267689	0,259856
10	0,257611	0,257653	0,256293	0,260668	0,261286

É visível pela tabela 5.19, que os tempos de execução resultaram em um desvio padrão de 0.012667 s no processo.

Tabela 5.19: Média total, máximo e mínimo no tempo de execução

Média total (s)	Máximo (s)	Mínimo (s)
0,259114	0,267689	0,255022

5.4.2 Tempo de aquisição de imagem pela câmara

O tempo de execução da câmara é fixo pelo *software*. Por ensaios práticos, definiu-se o uso de 3 segundos iniciais para que a câmara estabilize os parâmetros internos.

Neste caso, o ideal consiste em definir os parâmetros automáticos da câmara, para que não se tenha que "esperar" até a câmara os ajuste automaticamente.

O tempo atual de aquisição da câmara, representa um aumento de sensivelmente 12 vezes o tempo de execução de todo o código.

5.5 Tempo relativo de execução do código, se aplicado à placa de desenvolvimento *Raspberry Pi 3*

O código desenvolveu-se e executou-se no computador com as características apresentadas anteriormente, assim, deve ser feita a comparação de desempenho entre o *hardware* utilizado e o que se pretende implementar na construção do dispositivo móvel.

De forma a comparar o desempenho do computador utilizado, com o *Raspberry Pi 3*, utilizou-se o programa de testes designado por *Sysbench*. Este permite medir o tempo de execução de um determinado código, e apresenta o tempo de execução do mesmo. Pelo autor do ensaio [26], o *Raspberry Pi 3*, alcançou um resultado de 120,0822 segundos na execução do código. No entanto, o computador utilizado conseguiu um tempo de 1,1592 segundos, então o *Raspberry Pi 3* é 103,6 vezes mais lento. Como o código desenvolvido é executado em 0,259114 segundos, estima-se que no *Raspberry Pi 3* demoraria 26,8 segundos a completar a mesma execução.

Este tempo de execução é encorajador, pois, se o leitor estiver permanentemente ativo, permite executar sensivelmente 9720 leituras por mês. Um número de leituras muito acima das praticadas hoje em dia pelos contadores analógicos, instalados nas redes.

Capítulo 6

Conclusão e trabalho futuro

O problema apresentado nesta dissertação consistiu na elaboração de um sistema capaz de se adaptar a vários modelos de contadores analógicos, de modo a ser possível ler e transmitir os seus consumos de forma automática. Pretende também que, pela análise dos consumos efetuados, haja possibilidade de emitir alarmes que avisem o proprietário do contador sobre possíveis anomalias nos registos lidos.

Este estudo iniciou-se por uma revisão da literatura académica, onde se verificou que o procedimento de *retrofitting* pode ser realizado com base na aplicação de sistemas de visão, *encoders* e contagem por pulsos. Os dois últimos requerem aplicação de *hardware* específico para cada modelo do contador analógico. Contudo, a aplicação de sistemas de visão, é adaptável a vários modelos de contadores, tanto analógicos como digitais, e requer reduzidas alterações no contador em que se pretende aplicar o *retrofitting*.

Para a realização do protótipo desenvolvido para efetuar ensaios experimentais, utilizou-se um contador analógico de eletricidade. O protótipo é composto pelo sistema de leitura, pelo encapsulamento, câmara, iluminação artificial e controlo de rotação de contagem.

No *software*, necessitou-se de recorrer às bibliotecas de *OpenCV* e algoritmos *OCR*, que permitiram analisar as imagens da câmara de vídeo e possibilitou a extração do valor do contador. No lado do servidor, recorreu-se à ferramenta *Glade Interface Designer* para *interface* com o utilizador.

Assim sendo, o sistema é composto pelo leitor, responsável por ler o valor do contador, guardar os dados lidos, verificar possíveis fugas e enviar os consumos para o servidor. Este último tem a finalidade de receber os dados dos leitores e verificar fugas nas redes de abastecimento.

Nos resultados obtidos a nível de sistema de visão, verificou-se um aumento da taxa final de acerto, pela aplicação de abordagens diferentes desenvolvidas. Estas permitiram aumentar a taxa de acerto de 92% com 1000 imagens, para 100% em 8000 imagens. Este resultado deve-se à aplicação das ferramentas na análise de imagem e validação dos valores obtidos pelo algoritmo *OCR*. Na análise de imagem, o nível de visibilidade configurou-se a 93%, houve um aumento da resolução de 800 * 600 para 1280 * 960 pixels, que corresponde à resolução máxima da câmara e foram aplicadas as operações morfológicas da erosão e dilatação. Por outro lado, na validação de dados, configurou-se a *whitelist* e *blacklist* do algoritmo *OCR*.

Nos ensaios para deteção de fugas em habitações, definiu-se um limite de $0,1m^3/min$, no ensaio executado, verificou-se 37 passagens acima do limite máximo, num total de $17,3m^3$ de água desperdiçada.

Já no ensaio para deteção de fuga em redes de abastecimento, executou-se uma simulação com dois leitores. Neste ensaio, definiu-se um leitor com a leitura correta e outro com erro

associado. Os dados das diferenças entre os valores, resultou num valor que pode ser utilizado para deteção de fugas nas redes de abastecimento. O ensaio detetou de perdas de $1127,5m^3$ de água. É necessário referir que ambos os ensaios, foram simulados com dados do leitor, havendo assim uma necessidade de aplicar o sistema em ambiente real.

Relativamente ao tempo de aquisição de imagem pela câmara, é 12 vezes superior ao tempo de execução do código de análise da imagem. Isto acontece, porque a câmara necessita de ajustar os ganhos internos, e como tal, se não estiverem devidamente ajustados não é possível fazer a aquisição da imagem.

Como referido nos tópicos acima, os ensaios foram executados num computador, e não em placas de desenvolvimento. Todavia procedeu-se à comparação dos tempos de execução do código entre o computador utilizado e o *Raspberry Pi 3*. Concluiu-se que se a aplicação fosse executada por exemplo no *Raspberry Pi 3*, então uma leitura demoraria aproximadamente 27 segundos. O que significa que é possível executar acima de 9500 leituras por mês. Este número é relevante e encorajador para continuar o desenvolvimento do *software* e *hardware* para futuras aplicações práticas do sistema em diversos ambientes.

Futuramente, sugere-se análise aprofundada das opções disponibilizadas pelos algoritmos *OCR*. Não só o algoritmo *Tesseract*, onde algumas foram estudadas neste trabalho, mas também o *FineReader*.

Na iluminação artificial, é recomendado o uso de circuitos para controlar o sistema, e perceber a sua influência com na taxa final de acerto.

Igualmente é recomendado que seja estudado como os parâmetros internos da câmara influenciam da taxa de acerto e a velocidade de aquisição de imagem.

Como os ensaios das fugas foram simulados, é necessário testar o equipamento em ambientes reais, que podem ser habitações ou redes de abastecimento.

Por último, tratar e aprimorar a interface gráfica do sistema, bem como aplicar o desenvolvimento em página web, para melhorar o acesso não só pelo computador, mas também *tablets* ou *smartphones*.

Bibliografia

- [1] LUSA/SOL. *Rupturas e furtos de água custam 160 milhões por ano*. 2014. URL: <https://sol.sapo.pt/artigo/119673/rupturas-e-furtos-de-agua-custam-160-milhoes-por-ano> (acedido em 24/05/2017).
- [2] Offsite Data Backup. “MySQL Commands”. Em: *Trial* (2010), pp. 1–6. URL: <http://www.emacsa.es/atencion-al-cliente/individualizacion-de-suministros/>.
- [3] Cariátide. *Contadores digitais*. 2014. URL: <http://cariatideast.com/noticias/ver/12/index.html> (acedido em 24/05/2017).
- [4] Master Of, Science In e Electrical Engineering. “Automatic Utility Meter Reading”. Em: (2010).
- [5] Observador. *Relatório de Bruxelas responsabiliza fraude por derrapagem no défice da eletricidade – Observador*. 2014. URL: <http://observador.pt/2014/12/22/relatorio-de-bruxelas-responsabiliza-fraude-por-derrapagem-no-defice-da-eletricidade/> (acedido em 24/05/2017).
- [6] IEEE SPECTRUM. *Smart Meter Fires - IEEE Spectrum*. 2012. URL: <http://spectrum.ieee.org/energywise/energy/the-smarter-grid/smart-meter-fire-reports> (acedido em 24/05/2017).
- [7] European Commission. *Smart Metering deployment in the European Union | JRC Smart Electricity Systems and Interoperability*. 2017. URL: <http://ses.jrc.ec.europa.eu/smart-metering-deployment-european-union> (acedido em 24/05/2017).
- [8] European Commission. *Smart grids and meters - European Commission*. URL: <https://ec.europa.eu/energy/en/topics/markets-and-consumers/smart-grids-and-meters> (acedido em 24/05/2017).
- [9] Kestakon. *Siemens WFK40.D110, S55560-F105*. 2017. URL: <http://www.kestakon.co.uk/siemens-wfk40-d110-s55560-f105/> (acedido em 28/10/2017).
- [10] Proen Energetika Nekretnina. *Podžbukni ili ravni vodomjer*. 2017. URL: <https://www.proen.hr/podzbukni-ili-ravni-vodomjer/> (acedido em 28/10/2017).
- [11] Meters - WaterWorld. *Meters - WaterWorld*. URL: <http://www.waterworld.com/articles/print/volume-28/issue-2/products/products-services/meters.html> (acedido em 28/10/2017).
- [12] Zenner. *ZENNER Encoder Type Registers - Zenner*. URL: http://www.zennerusa.com/index.php?id_%product=52&controller=product (acedido em 28/10/2017).
- [13] WaterWorld. *Products & Services: Meters & Meter Reading Equipment - WaterWorld*. 2017. URL: <http://www.waterworld.com/articles/print/volume-29/issue-02/products/products-services-meters-meter-reading-equipment.html> (acedido em 28/10/2017).

- [14] Yachen Tang et al. “Extraction of Energy Information from Analog Meters Using Image Processing”. Em: *IEEE Transactions on Smart Grid* 6.4 (2015), pp. 2032–2040. ISSN: 19493053. DOI: 10.1109/TSG.2015.2388586.
- [15] K Aswathy e P P Narayanan. “Image Extraction-IE Approach to Extract Information from Analog and Digital KWh Meters”. Em: 6.5 (2016), pp. 6236–6240. DOI: 10.4010/2016.1505.
- [16] Miguel Rodríguez et al. “HD MR: A new algorithm for number recognition in electrical meters”. Em: *Turkish Journal of Electrical Engineering and Computer Sciences* 22.1 (2014), pp. 87–96. ISSN: 13000632. DOI: 10.3906/elk-1202-34.
- [17] Marcin Heliński, Miłosz Kmiecik e Tomasz Parkoła. “Report on the comparison of Tesseract and ABBYY FineReader OCR engines”. Tese de doutoramento. 2012, p. 24. URL: http://dl.psnc.pl/download/PSNC{_}Tesseract-FineReader-report.pdf.
- [18] Teliko. *Teliko*. URL: <http://www.teliko.com/en/remote-meter-reading/metlink>.
- [19] CAS Tecnologia. *Page not found - CAS Tecnologia*. URL: <http://www.cas-tecnologia.com.br/cas-hemera/42-solucoes-cas/305-linha-rs2000> (acedido em 24/05/2017).
- [20] Duodigit. *Modem DD3G-910 - DuodigitDuodigit*. URL: <http://duodigit.com.br/produtos/telemetria/modem-dd3g/> (acedido em 24/05/2017).
- [21] JANZ. *soluções de contagem*. URL: <http://www.janzce.pt/index.php?section=15>. (acedido em 24/05/2017).
- [22] Chamanstore. *CONTADOR DE MANO ANALOGICO MANUAL PARA GOLF, GOLPES, BOXEO, GIMNASIO, DEPORTES - Chamanstore.com*. URL: <http://www.chamanstore.com/acessorios-para-correr-running/67-contador-de-mano-analogico-manual-para-golf-golpes-boxeo-gimnasio-deportes.html> (acedido em 24/05/2017).
- [23] Osama Hunaidi. “Detecting Leaks in Water-Distribution Pipes”. Em: *Institute for Research in Construction* 92.October (2000), pp. 1–6.
- [24] DealExtreme. *WS2812 24-5050 RGB LED Driver Development Board - Black + White - Free Shipping - DealExtreme*. URL: <http://www.dx.com/p/ws2812-24-5050-rgb-led-driver-development-board-black-white-380044{\#}.WSbGjje1thE> (acedido em 25/05/2017).
- [25] Aliexpress. *Xbox Live Camera Reviews - Online Shopping Xbox Live Camera Reviews on Aliexpress.com | Alibaba Group*. URL: https://www.aliexpress.com/xbox-live-camera{_}reviews.html (acedido em 25/05/2017).
- [26] sysbench RaspberryPi. *(Tested) Raspberry Pi 3 vs Raspberry Pi 2: CPU and GPU Benchmarks (+ Burn-in Test) - Geeks3D*. 2016. URL: <http://www.geeks3d.com/20160322/tested-raspberry-pi-3-vs-raspberry-pi-2-cpu-and-gpu-benchmarks-burn-in-test/> (acedido em 27/10/2017).

Apêndice A

Principais aplicações instaladas

Tabela A.1: Principais bibliotecas instaladas

Sistemas de visão
<code>sudo apt-get install libopencv-dev</code>
<code>sudo apt-get install -y tesseract-ocr</code>
<code>sudo apt-get install libtesseract-dev</code>
<code>sudo apt-get install liblibleptonica-dev</code>
Interface Gráfica
<code>sudo apt-get install libgtk-3-dev</code>
<code>sudo apt-get install gtk-3-examples</code>
<code>sudo apt-get install glade</code>
<code>sudo apt-get install libcairo2-doc</code>
<code>sudo apt-get install glade-gtk2</code>
<code>sudo apt-get install gtk2.0 //glade versao2</code>
Gestão de ficheiros
<code>sudo apt-get install cmake</code>
<code>sudo apt-get install doxygen</code>
<code>sudo apt-get install doxygen-gui</code>
<code>sudo apt-get install graphviz</code>
<code>sudo apt-get install mysql-server</code>
<code>sudo apt-get install mysql-client</code>

Apêndice B

Lista das principais funções desenvolvidas no Cliente

Tabela B.1: Principais funções desenvolvidas

Funções destinadas ao Leitor
// Aquisição de imagem void ShowAndProcessWebcam();
// Aplicação da leitura
int CorteImg(void); int Leitura(void); void BlobsCleaner(void); int ThreshCallback(int, void*);
// Validação de dados
int MaisAltos(int num); int contaDigitos(int num); int validacaoNum(int num);
// Gestão de dados das leituras
void SaveLeituras(int num,int valor; int saveImage(int num,int Nleitura); void loadLeituras(void);
// Gestão dos ficheiros internos
void SaveData(const char *text); void SaveStatus(int coordenadaX, int coordenadaY, int distanciacumprimento, int distancialargura, int binlimitG;) void LoadStatus(void);
// Estatísticas e deteção de fuga
void FugaHabitacao(int num); void TesteImagem(); void Estatisticas();
// Programa Client int Client();

Apêndice C

Lista das principais funções desenvolvidas no Servidor

Tabela C.1: Principais funções desenvolvidas

```
//Funções detinadas ao SERVIDOR
```

```
void server(int porta);
void processclient(int sock, char *clientID);
void myerror(const char *msg);
void ManageCTRL_C(int dummy);
//funções utilizadas para o gtk
```

```
// Janela Help selecionar dispositivos
```

```
void bt_sair_help_select_dispositivos();
```

```
// Janela histórico de alarmes
```

```
void bt_historico_alarmes();
```

```
// Janela histórico de consumos
```

```
void bt_historico_consumos();
```

```
// Janela selecionar dispositivos da lista
```

```
void bt_select_dispositivos_lista();
void bt_help_selec_dispositivos();
void bt_sair_selec_dispositivos();
```

```
// Janela adicionar dispositivos
```

```
void combo_box_tipo(GtkWidget *widget);
void bt_conf_ad_dispositivos();
void bt_abrir_ad_dispositivos();
void bt_sair_ad_dispositivos();
```

```
// Janela de Help da aplicação
```

```
void bt_abrir_help();
void bt_sair_help();
```

```
// Janela principal da aplicação
```

```
void bt_sair_aplicacao();
void LeituraDados(void);
```

Apêndice D

Imagens da Interface gráfica



Figura D.1: Menu principal Servidor

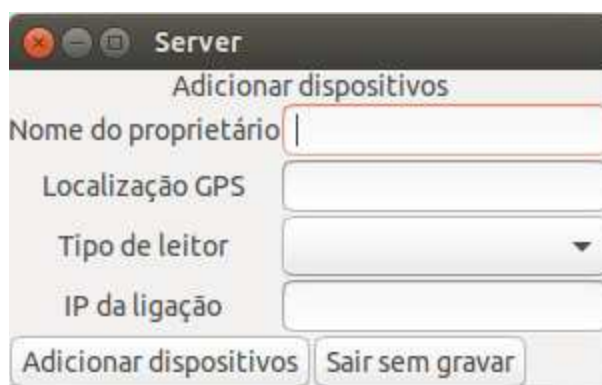


Figura D.2: Funcionalidade para adicional dispositivos

The image shows a screenshot of a software application window titled "Server". The window contains a form with the following elements:

- A header text: "Selecionar dispositivo para ver as suas características".
- A label "Selecionar dispositivo" followed by a text input field.
- A section header: "Ver dados do dispositivo selecionado".
- Four labels with corresponding text input fields:
 - "Nome do proprietário"
 - "Localização GPS"
 - "Tipo de leitor"
 - "IP da ligação"
- Two buttons at the bottom: "Sair" (left) and "Ajuda" (right).

Figura D.3: Funcionalidade para verificar as características